AD NUMBER
ADB346913
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to DoD and DoD contractors only; Specific Authority; 19 DEC 2008. Other requests shall be referred to Deputy Under Secretary of Defense (Science and Technology), 1777 N. Kent St., Suite 9030, Rosslyn, VA 22209.**
AUTHORITY
21 Oct 2009, per contributor, new PDF forwarded

Data Analysis Challenges

Contact: D. McMorrow - dmcmorrow@mitre.org

December 2008

JSR-08-142

JASON
The MITRE Corporation
7515 Colshire Drive
McLean, Virginia 22102-7539
(703) 983-6997

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
December 2008	Technical	

4. TITLE AND SUBTITLE

Data Analysis Challenges

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER
13089022

5e. TASK NUMBER
PS

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The MITRE Corporation
JASON Program Office
7515 Colshire Drive
McLean, Virginia 22102

8. PERFORMING ORGANIZATION REPORT NUMBER

JSR-08-142

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

OSD/DDR&E/DUSD (S&T)
1777 North Kent Street Suite 9030
Rosslyn, VA 22209

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

JASON was asked to recommend ways in which the DOD/IC can handle present and future sensor data in fundamentally different ways, taking into account both the state-of-the-art, the potential for advances in areas such as data structures, the shaping of sensor data for exploitation, as well as methodologies for data discovery. This report examines the challenges associated with the analysis of large data and in particular compares DOD/IC requirements to those of several data intensive fields. JASON finds that DOD/IC data requirements are certainly significant, but not unmanageable given the capabilities of current and projected storage technology. The key challenge will be to adequately empower the analyst by matching analysis needs to data delivery modalities. The report also proposes various grand challenges that could be used to assess and prioritize future research efforts in data assimilation and fusion.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			David Jakubek
Uncl	Uncl	Uncl	UL		19b. TELEPHONE NUMBER *(include area code)* 703-588-7412

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Contents

Abstract

JASON was asked to recommend ways in which the DOD/IC can handle present and future sensor data in fundamentally different ways, taking into account both the state-of-the-art, the potential for advances in areas such as data structures, the shaping of sensor data for exploitation, as well as methodologies for data discovery. This report examines the challenges associated with the analysis of large data and in particular compares DOD/IC requirements to those of several data intensive fields. JASON finds that DOD/IC data requirements are certainly significant, but not unmanageable given the capabilities of current and projected storage technology. The key challenge will be to adequately empower the analyst by matching analysis needs to data delivery modalities. The report also proposes various grand challenges that could be used to assess and prioritize future research efforts in data assimilation and fusion.

1 EXECUTIVE SUMMARY

This section summarizes the conclusions and recommendations of a 2008 JASON summer study commissioned by the Department of Defense (DOD) and the Intelligence Community (IC) on the emerging challenges of data analysis in the face of increasing capability of DOD/IC sensors. As the amount of data captured by these sensors grows, the difficulty in storing, analyzing, and fusing the sensor data becomes increasingly significant with the challenge being further complicated by the growing ubiquity of these sensors.

JASON was asked to recommend ways in which the DOD/IC can handle present and future sensor data in fundamentally different ways, taking into account both the state-of-the-art, the potential for advances in areas such as data structures, the shaping of sensor data for exploitation, as well as methodologies for data discovery. In particular, a salient question is the extent to which advances in the above areas can impact the central application of wide area surveillance. JASON was also asked to recommend assessment methodologies to both track progress and support future research; such methodologies could include the use of performance metrics, the implementation of test beds and the posing of competitions focused on grand challenge problems.

There is a perceived notion of a "capability gap" as regards future requirements for data management, with some forecasts predicting total data requirements in excess of a Yottabyte (10^{24} Bytes) by 2015 if current trends in sensor capability continue. These analyses are not credible in our view, in that they simply posit an increasing rate of data production without understanding the associated end-user requirements. It is of value to consider the evolution of data storage requirements arising from data-intensive work in scientific fields such as high energy physics or astronomy. Both these communities are faced with significant storage and analysis requirements, but

by matching the specific end requirements of their respective scientific goals, data filtering strategies have been developed, which in turn lead to more modest estimates for both storage and bandwidth. Typical data set size estimates for these communities will grow exponentially to a level of 100's of Petabytes by 2015.

Data volumes of this size are still very significant and do require specialized architectures and data analysis procedures. An examination of hardware trends in storage systems reveals that, despite exponential growth in the capacity of media over the past decades, it is becoming increasingly unlikely, absent the arrival of some disruptive technology, that this rate of growth can be sustained for single storage units such as disk drives. Instead, the high performance storage industry is applying distributed storage clustering approaches with great success. It is envisaged that technologies that can reliably store data sets of 100 Petabytes over time periods on the order of decades will be available in the near future.

JASON finds that similar conclusions hold for DOD/IC data analysis needs. The data requirements are certainly significant but not unmanageable given trends in storage technology. The key challenge is to empower the analyst by ensuring that results requiring rapid response are made available as quickly as possible while also insuring that more long term activities such as forensic analysis are adequately supported.

Requirements for the handling of data (particularly wide area surveillance data) will differ depending on timeliness requirements. Where time permits detailed retrospective analysis, JASON recommends the use of homogeneous data architectures, "cloud computing" (the provisioning of services from a generic cloud of servers) and the use of streaming data analysis algorithms that do not tie the data to particular data base schema or to a specific set of queries. Such approaches are currently in wide use by information providers such as Google and others. On more intermediate time scales, a service oriented architecture is appropriate and such applications are being

deployed by the DOD/IC. When rapid response is required, a push-based or event-driven architecture is most appropriate. For DOD/IC applications, the most critical metadata is accurate space and time registration. Combined with more accurate georegistration capabilities, this will more easily facilitate the analysis of correlated activity in locations of interest.

As the greatest challenge will come from the need to automate analysis, the most immediate need is for algorithmic advances that can help cue the analyst and trigger closer observation as well as possible fusing of other relevant data. The notion of fully automated analysis is today at best a distant reality, and for this reason, it is critical to invest in research to promote algorithmic advances; one way to effectively engage the relevant research communities is through the use of grand challenges in the area of data analysis. The key requirements for such grand challenges are that they focus on a difficult but ultimately achievable goal, be science-driven, and that success in such endeavors will leave a clear legacy in the target area. Several such challenges are suggested in the full report.

Our findings as regards data analysis challenges for the DOD/IC are as follows:

- DOD/IC data volumes as generated via various sensing modalities are, and will continue to be, significant, but they are in many ways comparable to those faced by other large enterprises.
- Important parallels can be drawn with data intensive science efforts such as high energy physics and astronomy.
- End user analysis requirements must drive the design of all aspects of the data enterprise including storage, database design and analysis tools.
- At present there is insufficient investment in software to more effectively process data as opposed to hardware to both collect and store data.

- Data organization and processing approaches such as cloud computing would appear to be best suited at present to facilitate future data fusion and discovery.
- Continued investment in technologies such as service-oriented architecture coupled with additional investment in event-driven architecture and software will be of benefit in enabling data fusion across the DOD/IC enterprise.
- Significant gains in data fusion can be realized in the short term through accurate spatial georegistration and time registration of sensor data.
- Processing closer to the sensor can yield important benefits provided there is a clear formulation of critical time sensitive data requirements.
- The greatest challenge will come from the need to perform automated analysis in support of the DOD/IC analyst.
- Grand challenges to stimulate further research in automated analysis can be used to assess and prioritize future research activities.

Given these findings, JASON recommends as follows:

- The DOD/IC communities should formulate a data analysis doctrine that
  - Continually assesses data requirements by matching analysis objectives to the data stream,
  - Focuses on homogeneous storage solutions with open interfaces,
  - Focuses on flexible analytic techniques that do not tie data to the query,
  - Focuses as strongly on software development as it does on sensor, storage, and network development, and

  - Differentiates between time sensitive analyses and retrospective analyses and applies the appropriate paradigm in each case.

- The DOD/IC communities should put into place efforts to validate the doctrine via several use cases.
- Continued investment should take place in interdisciplinary research in data analytics, machine learning and optimization.
- Invest in several grand challenges to assess and improve the state of the art in automated data analysis.

2 INTRODUCTION

This report describes the conclusions of a 2008 JASON study on data analysis challenges commissioned by the Department of Defense (DOD) and the Intelligence Community (IC). The focus of the study was on the emerging challenges of data analysis in the face of increasing capability of DOD/IC battle-space sensors. As the amount of data captured by these sensors grows, the difficulty in storing, analyzing, and fusing the sensor data becomes increasingly significant with the challenge being further complicated by the growing ubiquity of these sensors. For example, the DOD has developed and deployed a high resolution surveillance system called Constant Hawk. This system has the capability to capture synoptic data over a defined area. Current systems are capable of producing 10's to 100's of Terabytes [7] over a period of hours.

The difficulty faced in dealing with data at the volume generated by the Constant Hawk sensor is now typical of an emerging challenge. DOD missions now routinely exploit many high resolution sensors simultaneously (for example a swarm of UAV's) and must integrate multi-modal data. For some scenarios, short reaction times are critical, and so the relevant information must be delivered to analysts for decisions on short time scales. There is also a requirement that the information from the sensors be made available to a diverse community of users via a network.

JASON was asked to investigate and recommend ways in which the DOD and IC can handle this increasing volume of data in fundamentally different ways. We quote below the charge to JASON:

- Research the following areas of interest as far as evaluating which of these areas have the most promise of changing the way in which large data sets are handled:

Data architectures Both the size of the data to be transferred and the growing size of databases require novel architectural approaches to providing the adaptability and usability (automation and performance impact of human in the loop). Current databases, file systems, and network protocols will not keep pace. Which research areas and approaches have the most promise to impact DOD specific data challenges? Candidate research areas include reconfigurable scalable and dynamic systems; re-indexing, association and ontological representation for distributed and streaming data; many core file and operating systems, management and scheduling, and optimized algorithms; operationally relevant metrics and figures of merit for architectural performance, security and vulnerability.

Shaping sensor data for exploitation When tracing the processing chain from multi-source sensor inputs to the user/analysts, the techniques that are known and used become fewer and less mature. This simple process chain view goes from (1) metadata tagging to (2) preprocessing to (3) multi-source common data representation to (4) triage/identify high priority subsets for analysis and action. Candidate research areas include pattern analysis, data classification for importance and prioritization, criticality assessment, change detection, uncertainty management and reduction, high level structures, data search and retrieval, feature extraction, automatic translation, and automated or assisted pattern recognition.

Data discovery for exploitation In order to better discover and exploit the growing amount of sensor data, the following areas of research are considered: Object recognition in scenes and streams, discovery and exploitation at the edge, structuring knowledge for discovery, improving analytic throughput, aiding ISR functions, layered analysis and interpretation, effects prediction for decision support and cross domain access for effective ISR.

- Examine relevant DOD problem domains such as Wide Area Surveillance and Biometrics where recommended research areas can have an impact:

    1. What is the basis of making a decision or taking an action?
    2. How and why does this data make a difference?
    3. Is this a data feature that can be detected and processed (e.g. extracted) from the large data set? Do I know how, when and where to look for it within the large data?
    4. Is it possible to process the data to support the decision, action or analysis?
    5. Can it be done fast enough?

- Recommend assessment methodologies that will support the advancement of research to support solution development. Recommended areas to review include:

    1. What are the value metrics and/or performance metrics that can be standardized and used to compare research options and solution potential?
    2. What is the utility of standardized test data sets? Any specific recommendations?
    3. Any recommendations on the use of testbeds and experimentation?
    4. Recommend topics for a prize program to support data analysis or data exploitation

The study began with briefings from researchers from the DOD and DOE labs during the period June 23-27 who have been active in this area. We thank the briefers for their excellent presentations and in particular wish to acknowledge the work of David Jakubek who coordinated the briefings. JASON also extended invitations to several academics so as to get their

perspective as several research activities in areas such as high energy physics, astrophysics and climate modeling are now also facing this large data problem as their sensors become more capable. We also solicited the opinions of the computer science community as large data problems arise in many more quotidian settings such as search, data mining and other areas.

This report describes the summer study and the approach taken to respond to the DOD/IC charge. In Section 3 we discuss the nature of the large data challenge for DOD with an eye towards understanding the proper context for the problem. An important point of reference in this regard is related work in large data analysis being undertaken by several large science projects in the areas of high energy physics and astronomy. We argue in this Section that there is a fair amount of commonality between the large data analysis challenges faced by these communities and the DOD/IC and that future research efforts should take advantage of this commonality.

In Section 4 we examine the status of storage technologies for large data sets. The exponential rate of growth in storage capacity on single storage units has slowed in recent years indicating that, absent the advent of disruptive storage technologies, a distributed approach to storage is required similar in nature to what has happened for high performance computation.

In Section 5 we discuss how large data can be handled in different ways. We focus here on "schema neutral" approaches such as those utilized by information providers such as Google and Yahoo! which process data sets on the order of Petabytes routinely. These approaches make use of generic storage strategies, and significant use of redundancy. We argue that efficiencies can also be realized through the adoption of network based service oriented architecture for moderate to long time scale data interrogation problems requiring data source fusion while problems requiring rapid response will benefit from an event driven approach.

In Section 6 we examine several ideas for processing data closer to the sensor. It is not possible to stream down all sensor data from modern platforms and so some sort of on-board processing will be called for to send down limited subsets of data of interest. Such approaches include direct analysis of video streams and take advantage of new high performance commodity processors.

In Section 7 we examine the use of grand challenges as a mechanism for motivating research in areas associated with large data analysis that are relevant to the DOD/IC missions. We propose several ideas for problems that, if solved, can contribute technology that address the analysis needs most critical to DOD. Such problems could be posed as challenges to the research community with a prize offered for successful solution. This is similar in spirit to what has been done in the past with autonomous vehicles under the DARPA program. Finally, in Section 8 we summarize our findings and conclude with several recommendations.

3 DATA ANALYSIS CHALLENGES

In this section, we review the overall challenges facing the DOD as regards large data, but with the objective of putting these in the context of similar challenges facing other large enterprises. The difficulties facing the DOD are summarized in Figure 3-1 [12]. As can be seen in the Figure there is a notion of an emerging "capability gap". As the sensors associated with the various surveillance missions improve, the data volumes are increasing with a projection that sensor data volume could potentially increase to the level of Yottabytes (10^{24} Bytes) by 2015. At present, surveillance platforms such as the more recent Global Hawk system are capable of producing 10's to 100's of Terabytes [7] over a period of hours. In contrast, the Figure also shows that the capability of the DOD Global Information Grid (GIG) to transport or store this data is not keeping pace with projected growth. The concern is that much of the sensor data will not be processed. Indeed, the complaint has often been voiced in both the DOD and IC communities that even today "70 % of the data we collect is falling on the floor." [1] For reference, we provide a chart below that provides the prefix for the various powers of ten used to describe large data:

Power	**Prefix**
10^{9}	Giga
10^{12}	Tera
10^{15}	Peta
10^{18}	Exa
10^{21}	Zetta
10^{24}	Yotta

There are several difficulties with the projection in Figure 3-1. First, the projection simply posits the existence of future sensors (shown as Sensor X, Sensor Y, etc.) with ever increasing data outputs but with no clear connection as to the emerging technologies that will generate such outputs. To

[1] Quote attributed to Pete Rustan (DDR&E) at a MIT Lincoln Laboratory Senior Joint Advisory Council Review

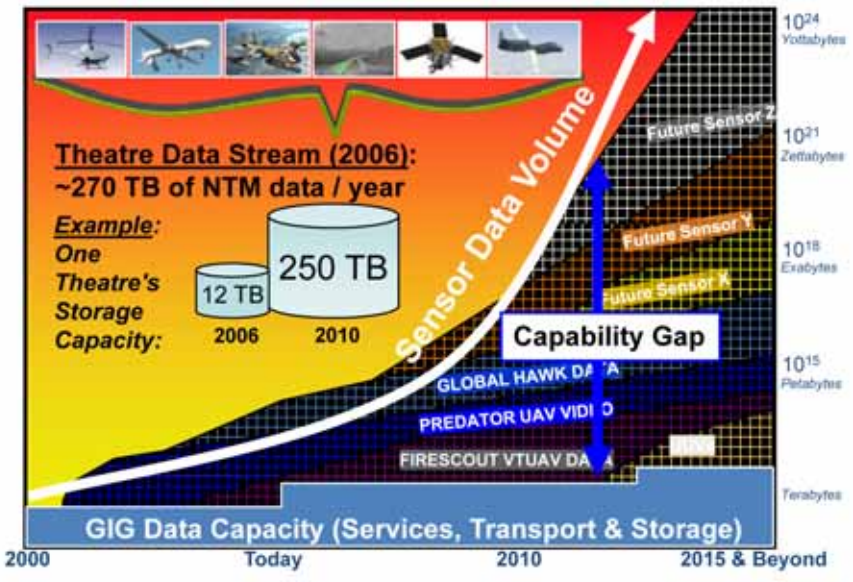

Figure 3-1: Projection of future data volumes for DOD sensor systems [12]

be sure, future sensor capabilities are improving and some discussion of near term capabilities is provided in Section 6, but absent the details, it is hard to know if such projections are valid. A more serious concern is the fact that the capabilities of various sensors are added one on top of another to create the ultimate projection of a Yottabyte of data by 2015. However, the graph is already in semilog form, meaning that adding capabilities on a log plot is equivalent to multiplying these capabilities together on a standard Cartesian plot. This means the projected growth is "super exponential". As we will see below, such growth is highly atypical if one compares the projections of Figure 3-1 to other data intensive enterprises. Finally, a plot of increasing data output has only limited utility as it assumes that all of the data is relevant to a given mission. In reality, data volume requirements will depend on the nature of the objectives.

To give some appreciation of the relative size of the data sets being considered in the extrapolation made in the capability gap diagram, it is instructive to understand the scale of a Yottabyte data set.[2] The earth has

[2]We are grateful to the JASON peer reviewer for this analogy and we quote from his

a surface area of 5.1×10^{14} m^2. If one images the entire surface of the earth (land, oceans, etc.) allocating one byte per square meter, that amounts to 0.5 Petabytes. If one were to image the entire surface of the earth with 1 m^2 resolution every second, after an hour one would accumulate 1.8 Exabytes. If one were to accumulate that data continuously for a month, one would have 1.3 Zettabytes. If one were to accumulate that data continuously for a year, one would have 16 Zettabytes. Finally, if one were to save an image of the earth at 1 m^2 resolution every second for 100 years, you would accumulate 1.6 Yottabytes.

The discussion above is not meant to imply that there is no challenge in handling and fusing the data that is currently routinely produced via surveillance. Indeed, DOD and IC data volumes are in many cases comparable to those encountered in other data intensive activities, particularly in data intensive science. It is instructive to examine two of these: high energy physics and the emerging field of synoptic astronomy as they represent use cases in which the response to large data volumes is connected to the ultimate scientific goals of the respective investigations.

3.1 The Case of High Energy Physics

The field of high energy physics is a key example of data intensive science. At a pedestrian level, a central goal here is to analyze the results of the collisions of high energy particles as a way of probing the fundamental forces of nature. As the accelerators used to explore high energy phenomena have increased in energy, these collisions result in an ever increasing profusion of collision products. At high energies, the problem is often akin to "finding a needle in a haystack" as the important events are hidden in a large background of other less physically interesting occurrences.

review of our report.

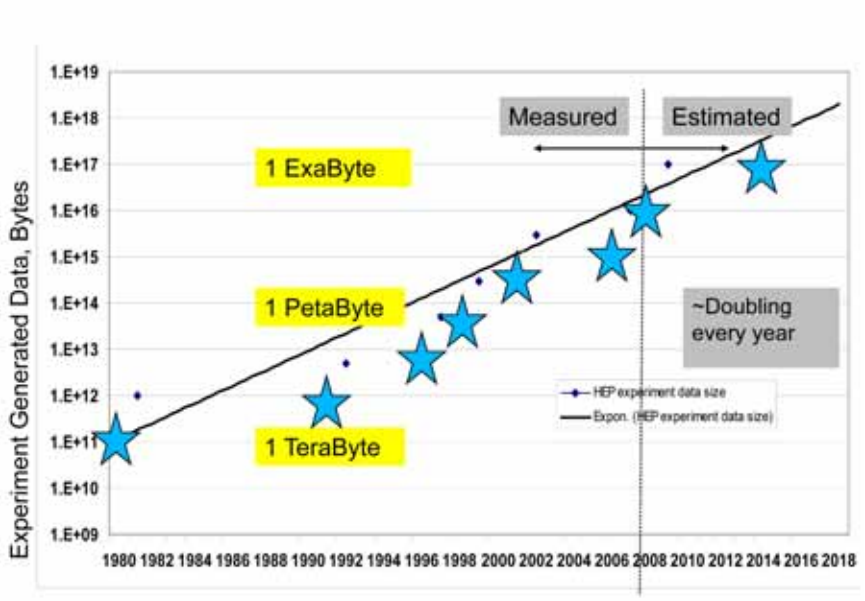

Figure 3-2: Data set sizes for high energy physics experiments plotted as a function of time. The small diamonds indicate the size of the experimental data sets. From about 1980 to the present the data growth is reasonably fit by an exponential which is also plotted as a guide to the eye. It is estimated that by 2015, the data set size will grow to be hundreds of Petabytes ($\sim 10^{17}$ bytes) [5].

As a result of the need to explore a large range of events, the data capabilites and requirements for various high energy experiments have increased rapidly over time [5]. The number of bytes generated in typical experiments is plotted as a function of time in Figure 3-2. As can be seen in the Figure, the growth is roughly exponential (not super exponential as indicated in Figure 3-1). If the projection is valid, the data requirements will be roughly 100's of Petabytes by 2015. It is anticipated that the storage capacity for such data sets will be readily available as will be further discussed below.

A key driver for this data increase is the Large Hadron Collider (LHC) which is now beginning to come on line at CERN in Geneva. The LHC utilizes an underground 27 km ring that was originally designed for electron-positron collisions to contain two opposing beams of protons that will be

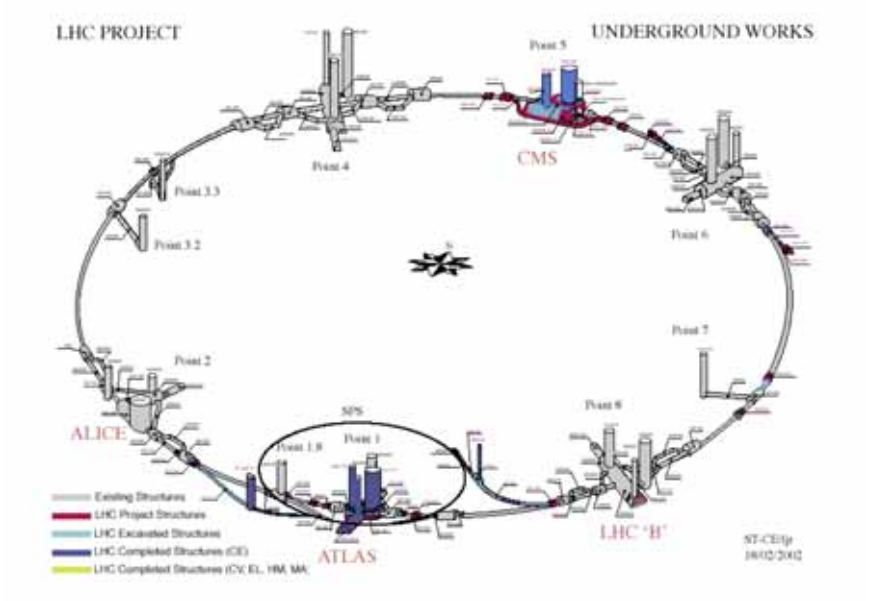

Figure 3-3: A diagram of the LHC [23]

made to collide with a beam energy of 7 TeV. A diagram showing the configuration of the LHC is provided in Figure 3-3.

The proton beams are actually bunches of protons with 2835 bunches of 10^{11} protons per beam. The bunch crossing rate when the protons collide is roughly 40 MHz and the collision rate is 10^9 Hz. The beams can be switched to various detectors as shown in Figure 3-3. The ATLAS and CMS detectors will be used to examine the results of the proton-proton collisions as part of the search for important new particles such as the Higgs boson [23].

A drawing of the ATLAS detector is shown in Figure 3-4. It is an enormous detector 46 meters long and 12 meters wide with a weight of 7000 tons. The 10^8 data channels available for recording data require on the order of 3000km of cables. It can be thought of as a very large sensor with the capacity to generate enormous quantities of data [21].

However, it is important to note that most of the events generated by the proton collisions will not be of interest. In fact, of the totality of the collisions,

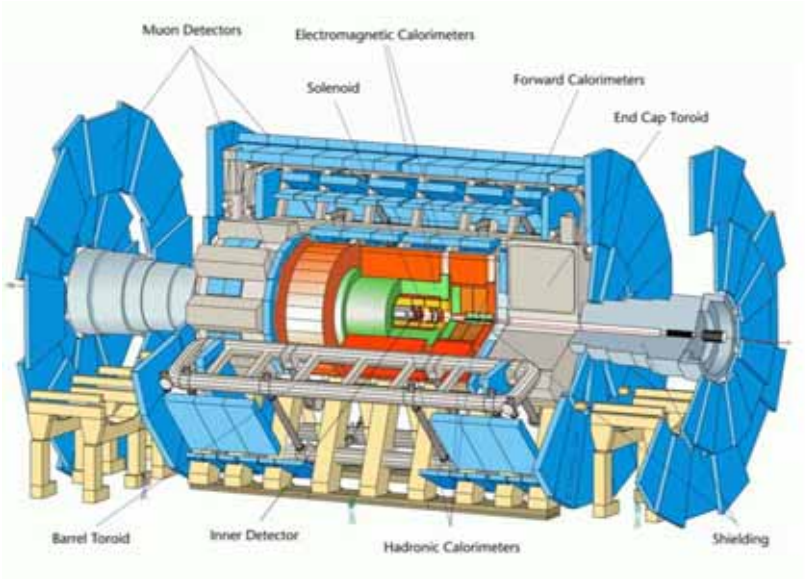

Figure 3-4: The Atlas detector [23]

the rate of events which are thought to possibly exhibit "new physics" is roughly 10^{-5} Hz and corresponds to an event selection rate of about 1 in 10^{13} [23]. In order to manage the potential data glut, much of it consisting of uninteresting events, the CMS and ATLAS detectors are set up to ignore the vast majority of events and to trigger only on those that are deemed interesting. The criteria for event tracking and recording are essentially built in to the experiment from the start. This represents therefore one extreme of the data analysis problem. Although an enormous amount of data can be generated, much of it is filtered allowing one to concentrate on events of interest. Of course one can argue that by doing this, some interesting events may be missed, but this is a compromise that is necessary in order to focus on a specific effect. In any case, this is an example where almost all of the data "falls on the floor", but because the data output are well matched to the expectation and interest of the relevant analysts, the approach has traditionally been successful.

Despite the filtering, the data requirements are still significant. Overall, even the filtered data will initially grow at the rate of tens of Petabytes per

year in the 2008 time frame and is expected to ultimately comprise thousands of Petabytes in less than 10 years according to initial estimates [3, 5].

Another important component of the LHC approach to their large data problem is the distributed nature of the collaboration. While the LHC is located at CERN, the LHC collaboration is international consisting of roughly 2500 physicists from 40 countries. Those events that are archived are then made available via an online store at CERN called "Tier 0". Tier 0 holds the raw data and also does processing to provide calibration data for further studies. Only a small subset of the collaboration has access to the full set of calibrations and reconstructions and access to the raw data is highly limited. A 10-40 Gbit per second network connects this central Tier 0 store to 10 Tier 1 sites around the world with the responsibility of reprocessing the full data with improved calibrations within two months of data taking. These analyses are then fed to a set of 30 Tier 2 sites also distributed around the world with the responsibility of production of simulated events. These Tier 2 sites are effectively the "physics caches". Finally, these analyses are made available to a larger set of Tier 3 sites which can perform interactive analyses on the simulated event data. This approach has the benefit of distributing responsibility in such a way that CERN's role is to generate the raw data along with the additional calibration needed to interpret it while the broad international community accesses and analyzes this data through its own hierarchical network. The main point is that the data storage and deployment is driven by the requirements of the experimenters and theoretical analysts. The overall approach is described graphically in Figures 3-5 and 3-6,

3.2 The Case of Synoptic Astronomy

The need for managing and fusing large sets of data also holds for the field of astronomy. Over time telescopes have become larger and, with the advent of multi-gigapixel cameras (in line with similar improvements in DOD

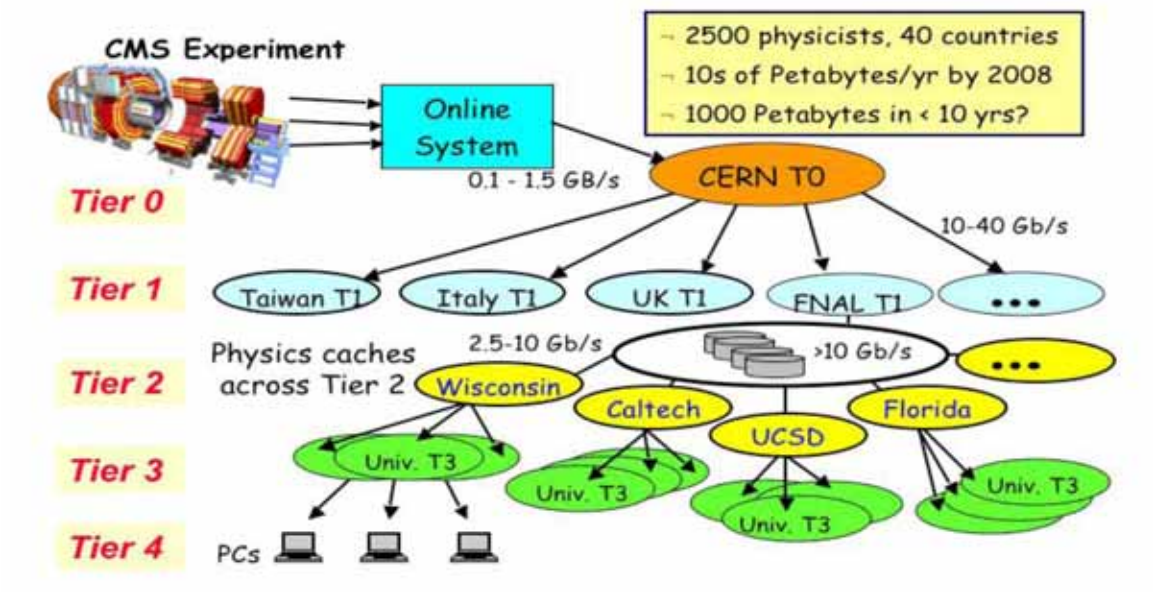

Figure 3-5: The LHC computing model uses a hierarchical networked approach to distribute data to collaborators based on their role in the project [5].

sensors), the field of astronomy must cope with the need to handle trillions of observations comprising collections of 50 or more Petabytes of data. The new paradigm is "synoptic" or "time-domain" astronomy, which involves constant refinement of the observations along with the ability to detect important time-dependent events such as supernovae or asteroids on a possible collision course with earth. This challenge has developed over time. The previous state of the art has been static surveys of the sky such as the Sloan Digital Sky Survey. However, in the near future projects such as the Large Synoptic Space telescope and the Pan-STARRS telescopes will image more of the sky more frequently. Here, one also looks for rare events as well as regular changes over time but, in contrast with the approach used by the LHC, all the data are archived. Given the size of the data sets and the rate with which they are generated, automated analysis is a key requirement.

As an example of the data sizes and rates we consider the Pan-STARRS telescopes which are now under construction on Haleakala in Hawaii. The

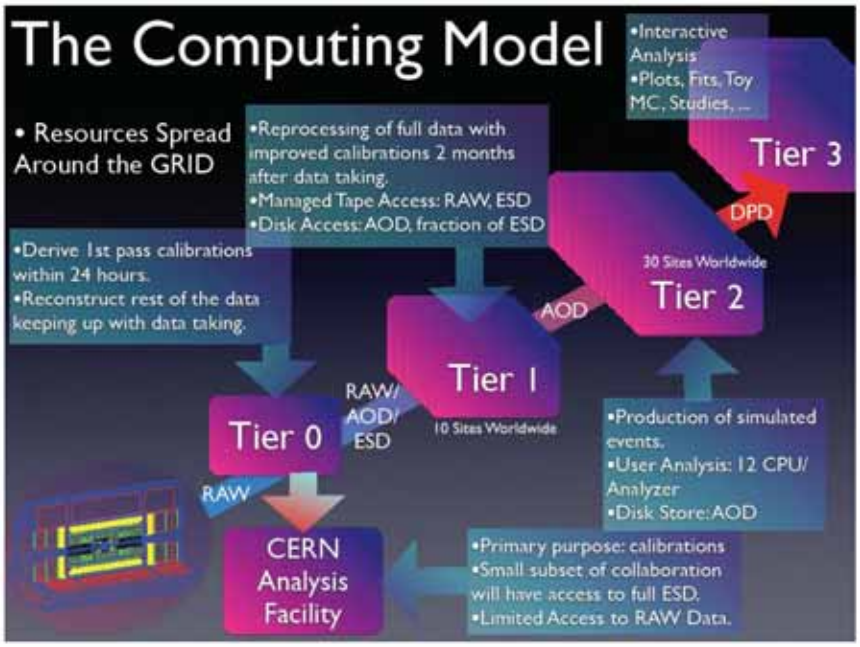

Figure 3-6: Functional decomposition of the tiered LHC computing model [23].

Pan-STARRS array (shown in Figure 3-7) will comprise four copies of the Pan-STARRS 1 prototype which utilizes one 1.4GPixel camera. It will provide 5 color imagery of 3/4 of the sky and is capable of making 12 visits to this part of the sky in 3 years. Pan-STARRS 1 by itself generates 2 Terabytes of data per night and a total of 800 TB per year. The complete system of telescopes is effectively a 4 by 1.4 Gpixel camera. The full array will provide 5 color imagery of 3/4 of the sky but will be able to generate 30 visits per year and generates 10 TB per night and about 4 PB in aggregate per year [25, 19].

The collection capability represents a significant shift in astronomy. It will be possible for example to constantly refine sections of the sky and update the collections as a result of the frequency of observation. In addition, the data can be used for change detection so as to identify fast "movers" such as asteroids. or other transients such as supernovae.

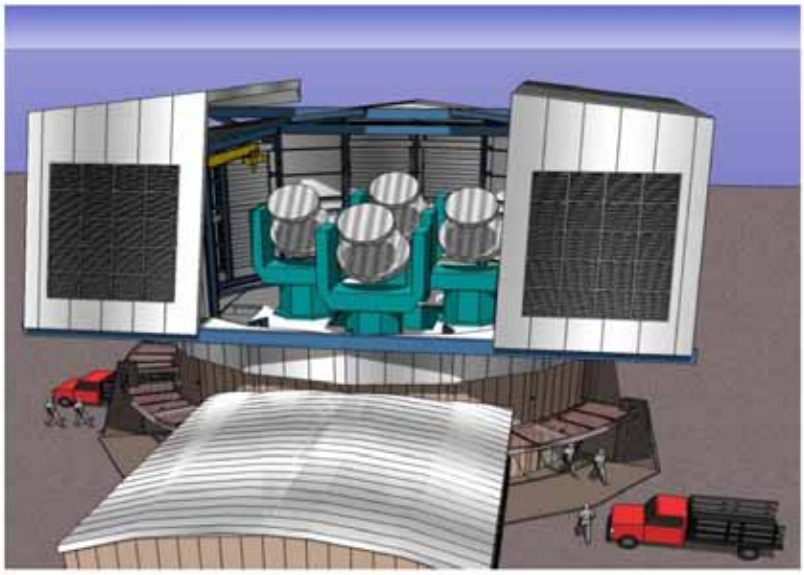

Figure 3-7: An illustration of the Pan-STARRS array [25].

In order to process the data, the Pan-STARRS project is developing an image processing pipeline utilizing essentially commodity storage solutions but which is well-matched to the needs of the astronomy "analyst" community. The data are served by a set of 80 "fat data bricks"(shown in Figure 3-8). Each brick will have 2 multicore processors with 16 GB of memory and 20 TB of disk using RAID 6 disk management. The entire system will serve 3 Petabytes for roughly $1M [25].

The Pan-STARRS data volume is certainly large but is very manageable given the capabilities of even commodity storage systems. For comparison, the Sloan Digital Sky Survey (SDSS) comprises 10 TB of images, and has 2-4 Terabyte catalogs of roughly 3×10^8 objects. Pan-STARRS will collect five colors and about 100 epochs for each pixel for a total of 10 Petabytes. This is comparable to Google Earth or Google Sky and about 100 times the size of the SDSS. By comparison, human capacity is more modest. All movie DVDs released to date comprise about a PetaByte and the text for all books ever published is "only" 30TB.

Figure 3-8: A storage element of the Pan-STARRS data pipeline. The storage uses only commodity components [25].

As we will discuss further in Section 4, the main issues in managing this volume of data are not rooted in hardware but in software. As we will show, there exist sound software approaches for collecting and curating the data making it possible to use commodity hardware to achieve the project requirements.

3.3 Data Requirements for Science and Industry

In light of the examples provided above for high energy physics and astronomy, it is also of interest to survey present day data requirements and data growth for a wider set of science experiments as well as the needs of those industries for which large data is a key aspect of their operations. Shown in Figure 3-9 are the rough data set sizes as a function of time for the BaBar high energy physics collaboration, the LHC discussed above, the data collections for NASA projects and the Large Synoptic Space Telescope (LSST). It can be

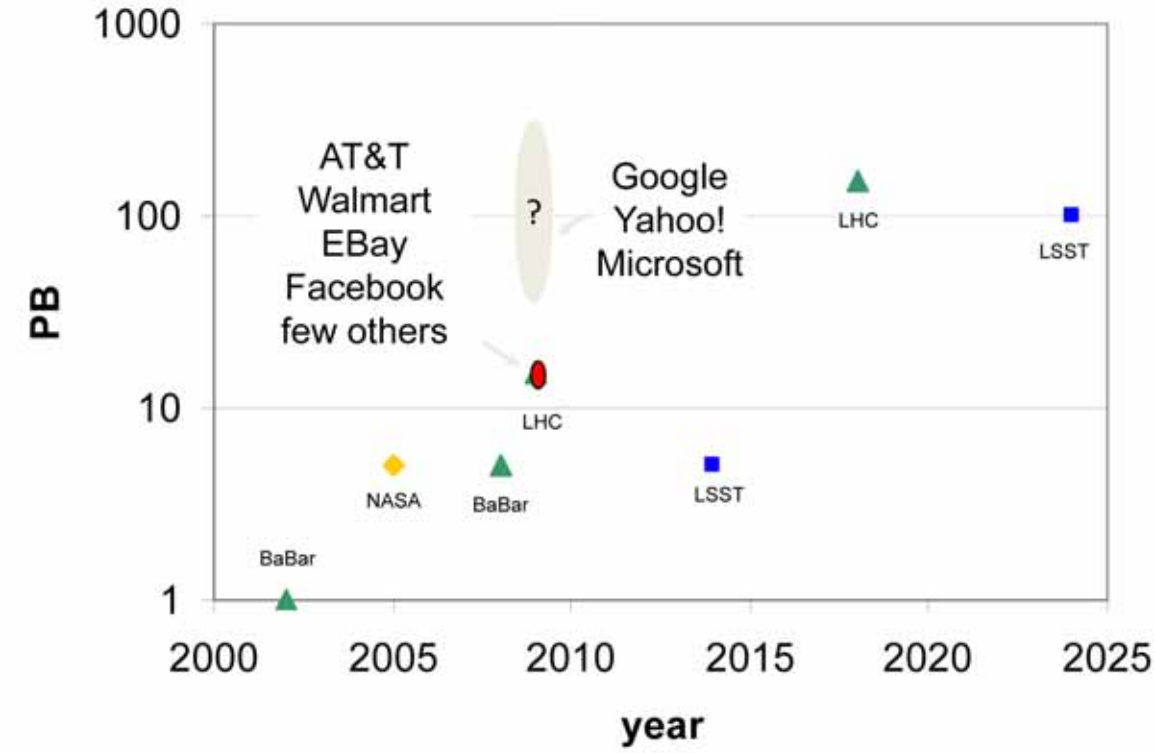

Figure 3-9: A plot of data growth (in Petabytes) for several data intensive science activates as a function of time. Shown also for comparison are data storage requirements for several corporations. Note that the rate of growth for the science projects is roughly exponential [3]

seen that data requirements for these efforts also rise roughly exponentially with time and would also seem to predict data volumes of roughly hundreds of Petabytes by 2020 [3].

The data requirements for industry are harder to gauge but there are several illustrative examples. Corporations such as AT&T, Walmart, EBay, Facebook and a few others serve on the order of tens of Petabytes. The data capabilities for truly data intensive businesses such as Yahoo!, Google, etc. are not publicly available but are estimated to be hundreds of Petabytes [3]

At least for these data intensive enterprises, there would not appear to be a case for serving Yottabytes of data at least on a 10 year horizon. The increase of data in other areas that utilize modern senor technologies such as

high energy physics and astronomy would seem to imply an exponential rise in requirements. This is not to minimize the need for state of the art storage technologies and it is of interest to understand if there are any hardware challenges to storing and manipulating this amount of data. We discuss this in the next section where we look at the development of modern storage systems and some of the issues that have arisen in light of the pervasiveness of data intensive applications.

4 STORAGE TECHNOLOGY

In this section we examine some of the trends in storage technologies that are relevant to the data challenges described in the previous chapter. We begin with a discussion of high performance I/O systems to uncover some of the technological challenges. We then indicate some of the possible solutions to these challenges. Interestingly, the trends show that there will be increased dependence on replication of data as well as increasing use of software to improve fault tolerance. Despite an anticipated increase in complexity, there is every indication that storage systems can keep up with the expected increase in data.

4.1 High Performance I/O Systems

The largest computers are used for scientific computation: large-scale simulation, climate modeling, weather prediction, petroleum, seismic, pharmacology, astrophysics. An example of a typical large installation is the Purple system at Lawrence Livermore National Lab (LLNL). The Purple system is a 1536 node parallel supercomputer capable of delivering 100 TFlops of computational capability. To support this level of computation, it provides a 2 Petabyte file system with a single mount point based on IBM's Global Parallel File System (GPFS). The particular file system configuration consists of 500 RAID controllers addressing 11,000 disk drives. The system can provide up to 126 GByte/sec to a single file and slightly more (134 GByte/sec) to multiple files. While this system will be exceeded in capability by future platforms, it is roughly representative of the state of the art.

The data analysis issues faced by the DOD differ from high performance computation in that most of the data comes from sensors, not data generated by a large scale computation. In that sense, it is a mismatch for some of the issues that are normally addressed in high performance file systems. There are be exceptions to this, such as large scale graphs created for applications such as network analysis.

High performance file system architectures are keeping pace with advances in storage technology and the requirements that are presented with each new generation of high performance computer system. The current generation file systems manage data on the order of 10^{16} bytes and the next generation systems will manage data on the order of 10^{18} bytes. These same file systems have I/O bandwidth on the order of 10^{11} bytes/second, and this is governed largely by the degree of parallelism available and the capacity of the interconnection network.

There are challenges that remain to be addressed. These include dealing with the complexity in the name-space that is introduced by the enormous capacity of these high performance file systems, and managing the vast archives of data that are produced by both simulation and data collection systems. Old paradigms for locating data based on a simple file path name break down when the number of files exceeds 10^9 as they now frequently do. Users have expressed the desire to locate data based on other properties of the data beyond its file name, including but not limited to its contents, its type and other semantic properties. Such a location service will require new indexing techniques that are currently subjects of academic research. We will discuss some of the approaches to these issues in Section 5.

The archival problem is significant, since there is an increasing desire to maintain copies of important data, but the creation rate of data will always exceed our ability to store it indefinitely. The Enterprise Storage Group estimates that by 2010 that the total digital archive capacity will be 25×10^{21} bytes. These trends are shown in Figure 4-1. The majority of this

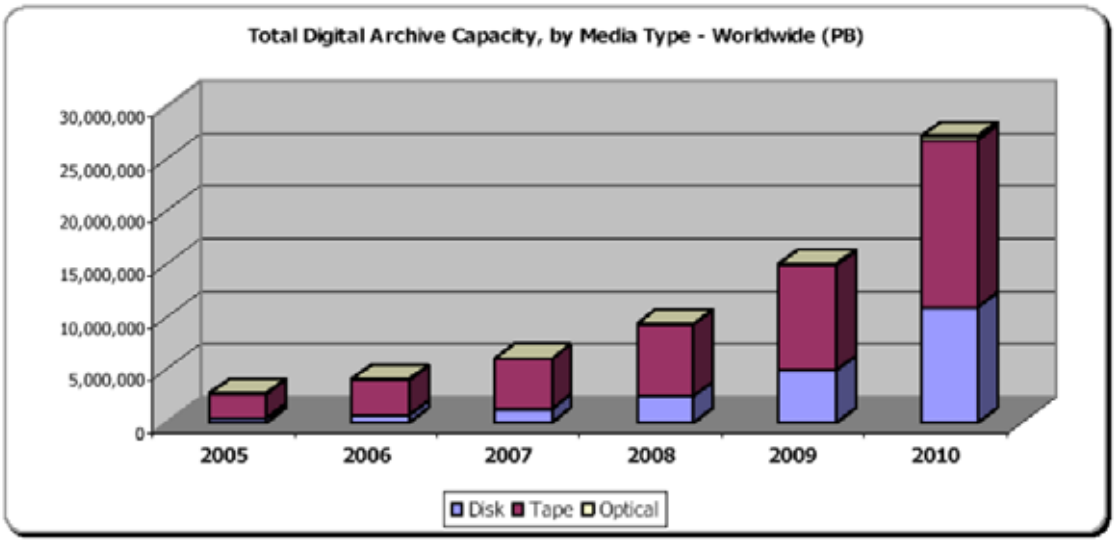

Figure 4-1: Estimate of the evolution of total digital archive capacity by media type

will be magnetic tape with the remainder being magnetic disk and only a negligible amount in optical storage. Magnetic disk is still too expensive to replace tape. There is research into making magnetic disk better suited for archives, including using less expensive consumer grade disks that are mostly turned off in order to manage the power budget.

The current generation of magnetic disks are 1.5Terabyte (Seagate 2008) and magnetic tapes are 1Terabyte/cartridge (IBM 2008) with a transfer rate of 160 GByte/second. There are significant differences in these technologies both in terms of cost and in performance. The cost of a reader for magnetic tape (currently $\approx \$39,000$ for the 1TB IBM system), is amortized over many tapes while each disk has its own reader and provides better performance (disks and tapes have comparable transfer rates, but since since disk has its own reader many disks can be used in parallel). Disks also have the advantage of providing random access, which is essential for many applications.

The result is that storage hierarchies remain important for performance and economic reasons. Since disk is faster, the upper level of the hierarchy

will be disk (there many be several levels of disk: fast enterprise class disks, and slower personal storage grade disks) and the lower level will be tape. Optical storage's low density and low bandwidth mean that it is not a serious contender (at least currently). Magnetic storage remains the highest density and highest performance for mass storage.

There are two performance metrics that are important to keep in mind: latency and bandwidth. Transferring large amounts of data is a bandwidth issue, and this is addressed through increased parallelism in both the I/O system and the interconnection network. The second issue is latency, and this is mainly an issue when performing many small operations, usually metadata look-ups. In many systems, the work of the file system is dominated by metadata operations. As with bandwidth, parallelism is the usual solution to latency issues, as well as heavy use of caching and clever data structures. Accessing the disk is approximately six orders of magnitude slower than main memory, so every effort is made to avoid disk accessed when possible.

Most parallel file systems strive to provide a familiar programming interface, and usually this interface is based on the POSIX specification. Additional layers such as MPI/IO are then added as middle-ware. Compliance with POSIX semantics puts constraints on the implementation of the file system, and if they are relaxed then significant performance gains can be realized.

4.2 Parallel File Systems

There are several efforts at building file systems for high performance computing. These include GPFS [22], Ceph [27], Lustre [4], and products such as Panasas [28]. In addition, there are numerous research efforts that are concerned with various issues related to high performance file system problems. The research can be found in a number of Computer Science re-

search conferences including: *Symposium on File and Storage Technologies* (FAST), *SC* (previously known as *Supercomputing*), *Symposium on Operating Systems Design and Implementation* (OSDI) to name just a few.

It is instructive to look at two extremes of the high performance file system design space. The first is GPFS, which operates as a clustered file system using a shared disk paradigm. The second is Ceph which, like Panasas, is a parallel file system that operates in the context of intelligent *object storage devices* (OSD) but is unique in completely avoiding traditional metadata approaches in favor of using pseudo-random data placement.

GPFS [22] was designed by IBM as a commercial parallel shared-disk file system that operates on the largest cluster computers in the world. GPFS attempts to the greatest extent possible to provide the same POSIX file system semantics as if it were running on a single computer instead of a cluster computer. It is the evolution of an effort known as *Tiger Shark* that was originally designed for streaming video servers. GPFS brings together a large number of ideas that were developed by the academic research community, in particular techniques for high performance locking and recovery. GPFS takes as its basic abstraction a distributed shared-disk architecture where all nodes in the cluster have uniform access to all the disks in the system.

An instance of the GPFS file system runs on a cluster of nodes. These nodes can also be used to run applications and may or may not have disks. Since GPFS provides a virtual disk abstraction, even nodes that lack disks access a disk interface presented by other nodes in the GPFS cluster. GPFS provides load balancing among the disks and over the interconnection network in an effort to provide the full throughput of the disk subsystem. The preferred configuration for GPFS is to use a switching fabric that directly connects file system nodes to disks: a storage area network (SAN), such as Infiniband, fibre channel or iSCSI. GPFS assumes a conventional block I/O interface with no intelligence at the disks.

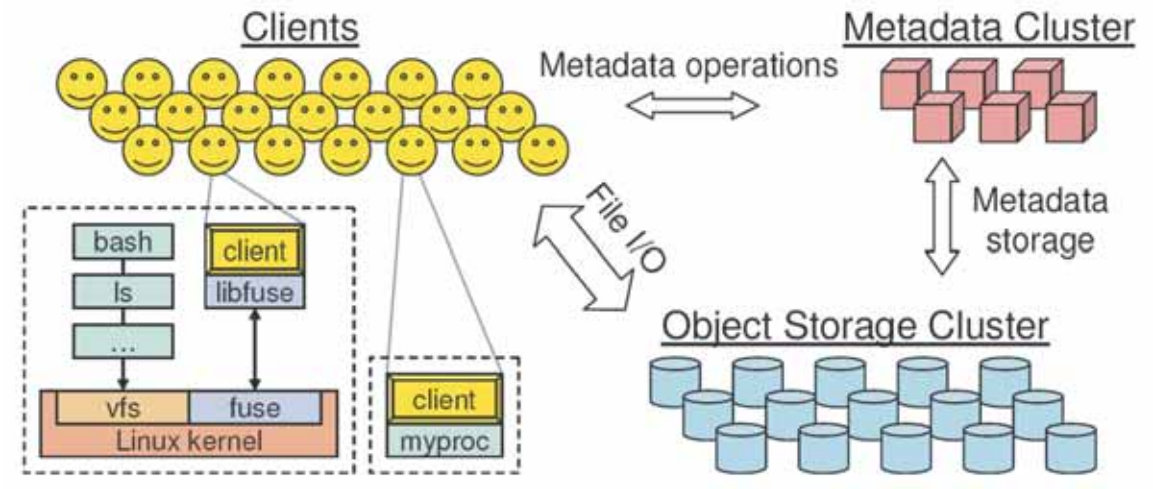

Figure 4-2: Ceph petascale file system architecture [27].

Ceph [27] is based on the assumption that systems at the petabyte scale are inherently dynamic: large systems are inevitably built incrementally, node failures are the norm rather than the exception, and the quality and character of workloads are constantly shifting over time. Ceph decouples data and metadata operations by eliminating file allocation tables and replacing them with generating functions. This allows Ceph to leverage the intelligence present in OSDs to distribute the complexity surrounding data access, update serialization, replication and reliability, failure detection, and recovery. Ceph utilizes a highly adaptive distributed metadata cluster architecture that dramatically improves the scalability of metadata access, and with it, the scalability of the entire system.

The Ceph file system has three main components: the client, each instance of which exposes a near-POSIX file system interface to a host or process; a cluster of OSDs, which collectively stores all data and metadata; and a metadata server cluster, which manages the name-space (file names and directories) while coordinating security, consistency and coherence (see Figure 4-2).

4.3 Technology Trends

It is of interest to examine recent trends in disk storage technology as they imply certain constraints for the engineering of storage systems of the future. As is well known, the exponential rise in computing speed for single processors has slowed over the original "Moore's law" estimate which posits that the number of transistors on a single chip roughly doubles every 18 months. This has in the past translated into a doubling of computing speed over the same time frame. However, due to difficulties in managing memory hierarchies as well as other issues, the rate of growth in computation speed has slowed. This is shown in Figure 4-3 which shows the evolution of the "specint" benchmark for commodity processors. The specint benchmark illustrates performance for integer intensive operations which are typical of image and other types of discrete analyses. As can be seen from the Figure, processing speeds have followed an exponential trend from 1985 to roughly 2003. After 2003, a "knee" develops in this curve indicating a slower rate of growth. Interestingly, a similar "knee" has been observed for disk storage systems.

Magnetic storage continues to hold the advantage in terms of density and cost, and this will continue to hold true for at least several years into the future. Eventually magnetic storage will reach fundamental physical limits, but in the past the technology has managed to side-step some of these limits by adopting new techniques. In Figure 4-4, the evolution of disk areal density is shown from 2000 through 2010. Areal density is the measure of the density of bits that can be recorded. Interestingly, until 2002 a rough doubling was seen. But after 2002 and through 2010 it is seen and anticipated that areal density for a single device will only grow at a rate of 35% to 45% per year. It was long believed that the limit on density would soon be achieved due to the super-paramagnetic limit on longitudinal recording. Magnetic disks have, for now, avoided this issue by moving to orthogonal recording. Similar

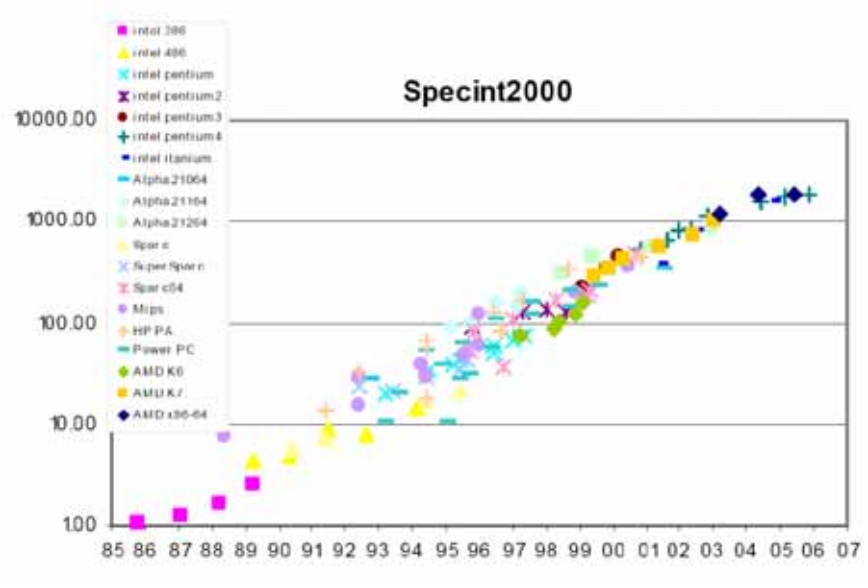

Figure 4-3: The specint benchmark over time for commodity processors

limits exist for orthogonal recording techniques, and so other approaches are being considered including heat assisted recording (a laser is used to heat a local area to increase the magnetic susceptibility), and the use of patterned media to create well-defined magnetic domains. The areal density in 2007 was 172 Gbyte/in^2 and is projected to grow to 472 Gbyte/in^2 2010. The growth rate in areal density has varied over the years: it was 62% in 1990–1998, an astounding 123% from 1998–2002, and fallen to 43% since 2002.

The other performance parameters of magnetic disks exhibit modest annual growth rates. Since magnetic disks are mechanical devices with complex control systems, physical limits related to power and manufacturing tolerances limit the potential for improvement. The access time (related to both seek time and rotational speed) has decreased by about 6% annually since 1990 from 12ms to 3.94ms today and is expected to reach 3.48ms in 2010. The seek time decreased by 7% annually from 1990 and by 4% since 1999 from 13ms to 3.52ms today and is expected to reach 3.19ms in 2010. This is summarized in Figure 4-5.

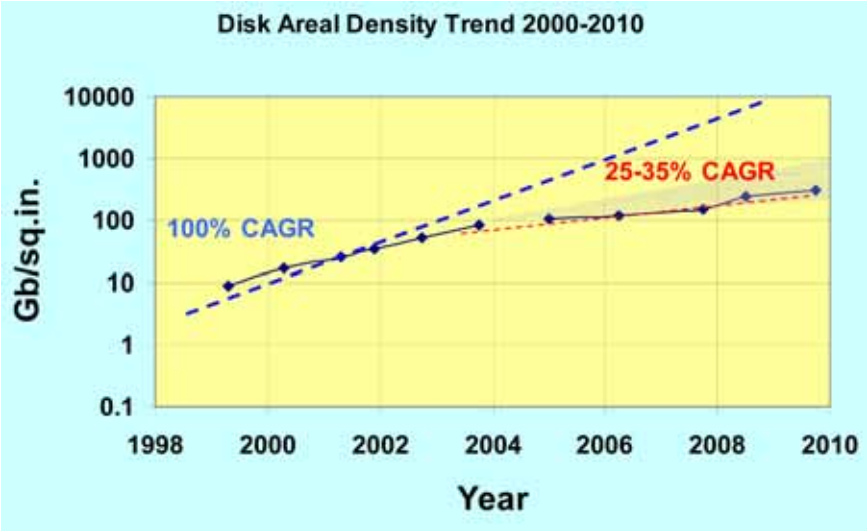

Figure 4-4: Areal density as a function of time for single disk drive systems [13].

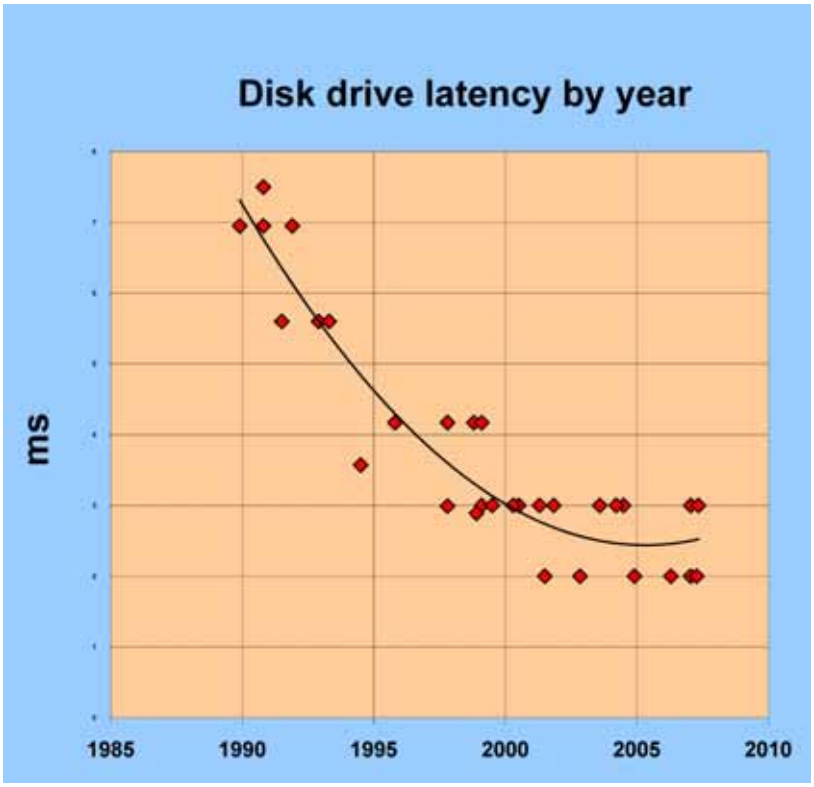

Figure 4-5: Disk drive latency as a function of time [13].

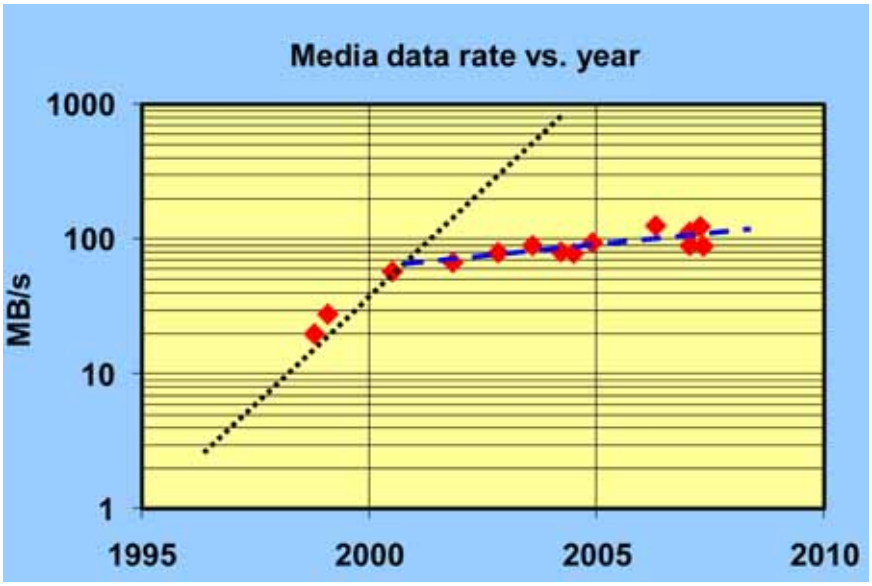

Figure 4-6: Data rate for disk drives as a function of time [13].

The sustained transfer rate grows roughly as the product of the rotational speed and the square root of the areal density (it comes from the track density, which is governed by the control system, and the linear bit density). Historical transfer rates are shown in Figure 4-6. A typical sustained transfer rate was 20MB/second in 1998, and is currently about 110MB/second. Again, it is clear that this is growing more slowly than prior to 2002.

Even though magnetic storage will remain the most economical storage technology in the near future, there are changes coming. Advances in flash memory mean that it is increasingly being used as an upper-level of the storage hierarchy. It has a much lower read latency than magnetic disk, but it is more expensive and has a lower density. As a result, some vendors are using flash memory in front of disk arrays as a way to improve performance, particularly for file system metadata operations.

There are other storage technologies on the horizon. There are typically grouped together under the name *Storage Class Memories* and include Phase Change or Ovonic memories, and Ferro-Electric RAM . Samsung has recently created a sample of a 512Mb device based on phase-change memory. Phase-

change memory uses the unique behavior of chalcogenide glass, which has different resistivity in its crystalline and amorphous states. These states can be changed by varying the heating and cooling profile of the memory cell. The difference in resistivity is several orders of magnitude, and so recent research has focused on establishing several levels and using this to store multiple bits per cell.

Resistive random-access memory (RRAM) is being developed by a number of companies. Different forms of RRAM have been disclosed, based on different dielectric materials, spanning from perovskites to transition metal oxides to chalcogenides. The basic idea is that a dielectric, which is normally insulating, can be made to conduct after application of a sufficiently high voltage. The conduction path may be reset by the application of a suitable voltage. Hewlett-Packard announced in April 2008 the development of a memristor, a previously unrealized circuit element that is another possible demonstration of RRAM, and subsequently announced they would begin prototyping RRAM using their memristors. There are also more speculative technologies such as carbon nanotubes that hold promise but have proven extremely difficult to take from laboratory demonstrations to a manufacturable product.

It had been hoped that MRAM or magnetoresistive random access memory, which uses the magneto-resistive effect on small magnetic domains, would provide a fast and dense non-volatile memory technology, but it now appears that it is reaching fundamental limits of cell size that are larger than existing flash cells. It seems destined to remain a niche technology, mainly of interest because it is radiation-hard.

Flash memory is by far the most popular non-volatile solid state memory technology, but it suffers from many limitations that make it poorly suited for large-scale data storage. Flash memory is limited by the duration of the erase/write cycle (which takes a very long time) and by its durability due to the limited number of erase/write cycles (on the order of 10^5).

Most commercially available flash products are guaranteed to withstand 10^5 erase/write cycles for block 0 but provide no guarantees for the other blocks. Recent research may extend the number of write cycles using ferroelectric NAND flash (the inventors claim 10^8 erase/write cycles). As a result of the limited number of erase/write cycles, wear leveling is required. This involves heuristic algorithms that attempt to place data that is unlikely to be written on blocks of the flash memory that are more worn (endured more erase/write cycles) while placing data that is actively being written on less work blocks. There is some degradation due to reads as well since a small amount of charge is lost each time, but the number of read cycles before an error occurs is large and the reports are anecdotal.

The density of flash also appears to be approaching a fundamental limit, since the number of electrons stored in the gate is already small (some engineers joke that there are so few electrons that they have names). Multiple bits per cell are stored by having different levels of charge stored, but this decreases the reliability of the memory.

Another interesting technology is holographic storage. The idea here is to encode data in the form of holograms in several layers of a photosensitive storage medium and to then read these back at high data rate using CMOS detector technology. InPhase technologies [24] has developed a "write once read many"(WORM) holographic disk storage system which shows promise for archival data although at present data densities and transfer rates are comparable to magnetic storage approaches. There have been further technology demonstrations by InPhase that achieve significantly higher density with promising results and so over time it may be possible to achieve storage volumes on the order of Petabytes from a single storage unit but this technology appears to be several years in the future.

The overall conclusion that one can draw from the considerations above is that the pace of improvement in storage technology has for the most part slowed. Current trends indicate we cannot expect single storage units that

can handle an exabyte or even a petabyte in the near future. Barring the development of some disruptive technology (such as holographic storage as discussed above), the only way forward is to use storage devices in parallel. This is already being done for large scale parallel computing systems partly to maintain data transfer rates and for redundancy in storing valuable data. It is a key component of the data handling strategy for the research areas discussed in Section 3. In the short term, this approach is the only direct way to achieving large scale storage. The use of many storage devices in parallel requires the consideration of hardware failure rates for components of storage systems and this is described in the following section.

4.4 Estimates of HDD Reliability

The most commonly cited measure of *hard disk drive* (HDD) reliability is *mean time between failures* (MTBF), which manufacturers currently give in the range of $350,000$ to $1,200,000$ hours over the lifetime of a population of enterprise class HDDs. It is important to remember that this measure is over the entire population of that HDD family and not for the individual HDD. The correct interpretation is that the entire population is expected to accumulate the MTBF of operational hours before it begins to experience failures. It is more realistic to consider the *service life* of the HDD or the *warranty period,* since in the later case these are the periods of time the manufacturer expects a minimum number of failures based on an economic analysis.

There are also significant problems with the estimation of the MTBF, since these numbers are often derived from sparse data sets and based on assumptions such as constant failure rates. Manufacturers will, for example, often calculate the MTBF taking the reciprocal of the calculated repair rate, ignoring the infant mortality of the HDD population.

It is commonly assumed that both HDD and RAID system failures follow a homogeneous Poisson process. The calculation of meant time to failure (MTTF) and mean time between failures (MTBF) are based on the exponential assumption, which has been show to be untrue [10]. This assumption is also used in the case of RAID arrays to derive *mean-time-to-data-loss* (MTTDL) also known as the average time to *double-disk failures* (DDF). Even if the HDD were to follow a homogeneous Poisson process, and it has been shown that they do not, there is no statistical reason to believe that the RAID system as a whole will follow such a process. It has been shown that inaccurate modeling using MTTDL which assumes an constant failure rate and an exponential distribution underestimates the number of double disk failures by a factor of 2 to 1500 times [9]. The current state of the art for modeling HDD reliability is to model the hazard rates as they evolve over time by fitting multiple Weibull distributions [8, 9].

There is a misconception that SCSI (enterprise class) HDDs and ATA (personal storage class) HDDs are internally the same technology and only differ in the external interface [1]. In fact they differ significantly in performance, reliability, and failure modes. These differences are driven by various factors, including cost pressure and desired performance. Enterprise class HDDs are typically higher performance with more rapid seeks, higher rotational speeds and reliability; personal storage class drives are typically slower, less reliable, but have a higher areal density since cost per unit capacity is important in that market. RAID arrays made of many personal storage class HDDs are less expensive, but care must be taken to deal with temperature and vibration, while enterprise class HDDs are engineered for this environment.

It is also important to keep in mind that several factors can lead to reads which are correct with regards to the error correcting codes (ECC) , but are erroneous. Perhaps the most obvious is a burst of errors so large that it exceeds not only the ability of the ECC to correct the error, but its ability to detect the error. More common are erroneous reads which come

Table 4.1: Range of read error rates [9].

Read Errors per per Byte per HDD		Low Rate 1.35×10^9	High Rate 1.35×10^{10}	
Low	8.0×10^{-15}	1.08×10^{-5}	1.08×10^{-4}	
Medium	8.0×10^{-14}	1.08×10^{-4}	1.08×10^{-3}	Errors/Hour
High	3.2×10^{-13}	4.32×10^{-4}	4.32×10^{-3}	

from servo tracking errors: two writes occur next to each other along the track, and depending on the alignment of the read head one or the other will be read.

Table 4.1 presents data on observed error rates. A study of $282,000$ HDDs by Network Appliance in 2004 found a *read error rate* (RER) of 8×10^{-14} errors per byte read. Other analyses have found RER of 3.2×10^{-13} errors per byte read among $66,800$ HDDs and a study of $63,000$ HDDs over five months found an RER of 8×10^{-15} errors per byte read. While it is possible using current technology to read 4.32×10^{12} bytes/HDD/day, the study of $63,000$ previously mentioned had an average read rate of 2.7×10^{11} bytes/HDD/day. While recognizing that we are working with averages, if we take the middle values then for 100 HDDs we can expect to have a read error approximately once per month.

4.5 Interconnection Network Failure

The availability of large-capacity, low-cost storage devices have led to active research in design of large-scale storage systems built from commodity devices for super-computing applications. Such storage systems, composed of thousands of storage devices, must provide high system bandwidth and exascale data storage. A robust network interconnection is essential to achieve high bandwidth, low latency, and reliable delivery during data trans-

fers. However, failures, such as temporary link outages and node crashes, are inevitable. It has been shown [29] that a good interconnect topology is essential to fault-tolerance of a exascale storage system.

System architects are building ever-larger data storage systems to keep up with the ever-increasing demands of bandwidth and capacity for supercomputing applications. While high parallelism is attractive in boosting system performance, component failures are now the rule rather than the exception. In an exascale storage system with thousands of nodes and a complicated interconnect structure, robust network interconnection is essential but difficult to achieve. Transient failures will be common.

Failures, which appear in various modes, have several effects on a large-scale storage system. The first is connectivity loss: requests or data packets from a server may not be delivered to a specific storage device in the presence of link or switch failures. The result is disastrous: many I/O requests will be blocked. Fortunately, today's storage systems include various levels of redundancy to tolerate failures and ensure robust connectivity. The second effect is bandwidth congestion caused by I/O request detouring. The average size of a single I/O request can be as large as several megabytes. Suppose that such a large system suffers a failure on a link or delivery path on an I/O stream. In this case, the stream has to find a detour or come to a temporary standstill. The rerouting will bring I/O delays and bandwidth congestion and might even interrupt data transfer. The I/O patterns particular to high-performance computing demand a network architecture that provides ultra-fast bandwidth and strong robustness simultaneously. The third effect is data loss caused by the failure of a storage device. As disk capacity increases faster than device bandwidth, the time to write and hence to restore a complete disk grows longer and longer. At the same time, the probability of single and multiple failure increases with the number of devices in the storage system.

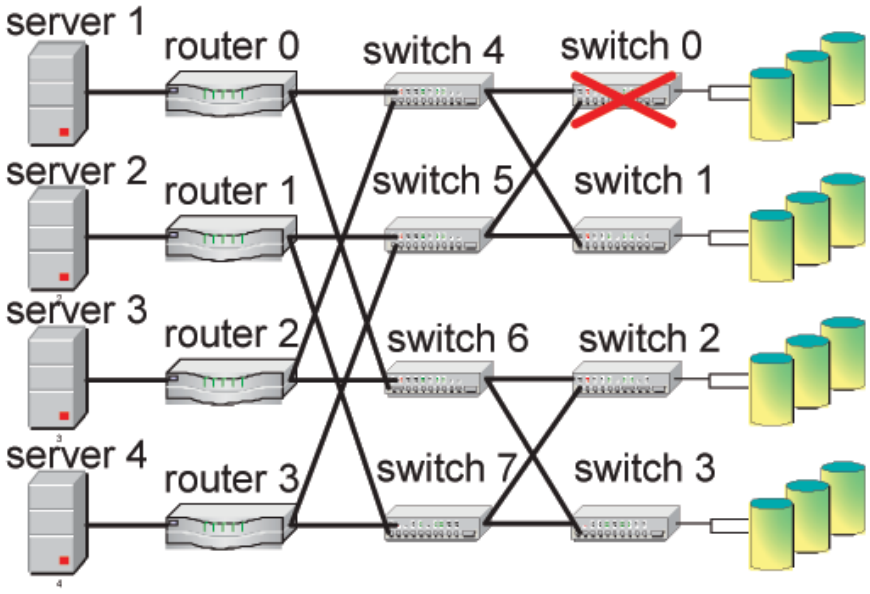

Figure 4-7: Butterfly networks under failures.

There are three primary failure scenarios to consider: link failure, connection node failure, and storage device failure.

1. *Link failure*: The connection between any two components in a system can be lost. If there is only one path between two components, a system is at risk when any link along this single path is broken. A robust network interconnection must be tolerant of link failures. Multiple paths between two components will decrease the vulnerability of a single-point of failure and effectively balance I/O workload.

2. *Connection node failure*: Connection nodes include switches, routers, and concentrators that link servers to storage nodes. They are used for communications and do not store any data. Compared with link outage, failures on an entire switch or router are more harmful for network connection since a number of links that were attached on the switch or the router are simultaneously broken, but losing connection nodes will not directly lead to data loss.

3. *Storage device failure*: When a storage device fails, it cannot carry any load. Further, additional traffic for data reconstruction will be generated. The increase in bandwidth utilization brought by data construction is of great concern when data is widely declustered in such a system.

4.6 Approaches to Enhanced Storage System Reliability

The sections above detail the challenges of developing large scale storage systems. An additional complication is that given the distributed nature of the DOD mission, we can expect that storage systems will also be distributed over multiple locations. Indeed, this is the concept of grid computing and storage. Access of data across a grid presents several challenges. There are several approaches:

Explicit copying This is the simplest approach and is exemplified by protocols such as ftp or Gridftp. The issue here is that keeping track of multiple copies of the data is tedious and error-prone. It is also difficult to maintain data provenance which as we will discuss later is essential. Finally the scheduling and planning of data management and synchronization are logistically very challenging.

Replica management This approach is exemplified by approaches such as Globus RLS. It requires global registration of managed storage objects but more importantly, it requires that data replicas be kept in sync manually or via separate tools.

File access protocols Here the picture is one of one copy of the data with updates done directly to a server (as in NFS). This is not scalable as it requires high bandwidth and low latency and offers little or no parallelism.

System	Year	TF	GB/s	Nodes	Cores	Storage	Disks
Blue Pacific	**1998**	**3**	**3**	**1464**	**5856**	**43 TB**	**5040**
White	**2000**	**12**	**9**	**512**	**8192**	**147 TB**	**8064**
Purple/C	**2005**	**100**	**122**	**1536**	**12288**	**2000 TB**	**11000**
HPCS (rough est.)	**2011**	**6000**	**6000**	**65536**	**512K**	**120000+ TB**	**200000+**

Figure 4-8: Computing and storage requirements for several existing high performance computing systems as well as the future DARPA high productivity computing system (HPCS)

A natural solution is to use a cluster of parallel file systems such as GPFS. In fact, this is currently deployed at major sites of the NSF TeraGrid offering a 500 GByte shared file system over a 30GByte per second backbone. While this does work and eliminates the need for multiple copies of data, the disk throughput is limited by network bandwidth and, more critically, if a portion of the network goes out data becomes unavailable.

Given the discussion above regarding storage system reliability, modern computers continue to push the growth rate of Moore's law through increasing parallelism. Some characteristics of current and future computer and associated storage systems are shown in Figure 4-8. As can be seen, future systems will require hundreds of thousands of computational cores as well as disks to meet the requirements of maintaining Moore's law in the face of flattening capabilities for processors and storage media. These challenges are pushing storage providers to develop global peer to peer file systems. The idea here is that a file spans multiple sites and is also replicated across those sites. This allows the application of traditional ideas like caching where data is moved into position so as to be ready for use but to be reread should the data change.

In this picture, the I/O nodes of the storage system become much more sophisticated and must participate in cache management as well as error correction. The requirements for future file systems and storage are substantial. For the file system, one requires balanced capacity and performance. For the applications discussed earlier, one expects something like a 100 Petabyte file system with a file I/O rate of something like 6 TByte/sec. The system will need to be reliable in the presence of localized failures. At the scales considered here, one or more of the drives will continually be in a state of rebuild given the error rates for drives discussed earlier. The rebuild overhead must be at an acceptable level. Standard RAID arrays are not appropriate for this purpose. RAID rebuilds can severely affect performance as the data is not available anywhere else. In general, for traditional parallel file systems, an $x\%$ degradation in service on one Logical Unit (LUN) of the file system will translate into a similar degradation across the entire file system.

One solution as briefed to us by Haskin [13] is to dispense with hardware RAID controllers and instead employ a more sophisticated I/O node. In this case the RAID function would be performed in software using much stronger error correcting codes so as to ensure longer mean time to data loss (MTTDL). IBM has proposed the use of Reed-Solomon codes that can ensure an MTTDL of 10^5 years for a 100 PetaByte file system. Additional safeguards include the use of end-to-end disk to file system to client check sums to ensure that data does not get silently corrupted, and the use of declustered RAID so that the rebuild and repair operations can take place with minimal ($\sim 2\%$) performance degradation.

The latter idea is very much in the spirit of distributed large scale file systems as we discuss in the next Section. In a conventional partitioned RAID, one partitions the drives into arrays and then creates LUNs on top of these arrays. As a result, one can only add drives in quanta of one partition. A rebuild operation will take place on the remaining drives of a given array. This is shown on the left of Figure 4-9 along with the relative read and write throughput required for a rebuild. Because of the way the data are organized

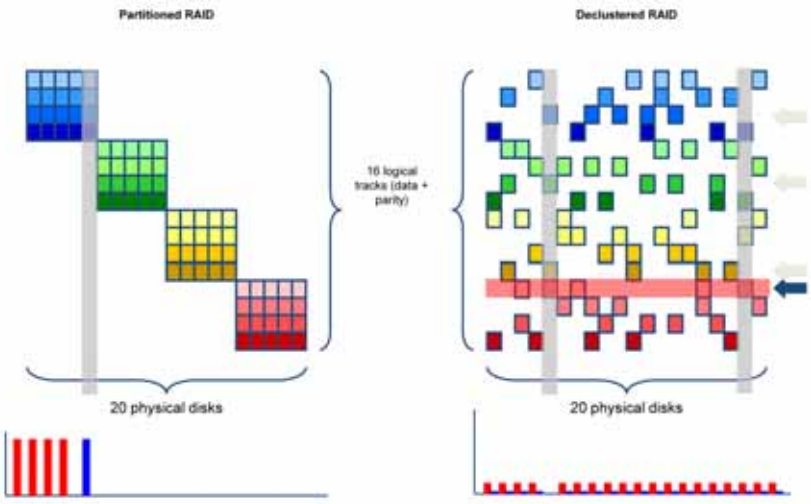

Figure 4-9: Read and write throughput associated with a RAID rebuild on a conventional partitioned RAID (left) vs a declustered RAID array

this makes the rebuild operation quite significant. In contrast, declustered RAID distributes data and parity strips of logical tracks evenly across all drives. This allows for an arbitrary number of drives in the array and individual drives can then be added and removed as necessary. In addition, the cost of rebuild is then spread evenly over the entire array.

Using the ideas described above it is possible to construct file systems with a high level of reliability and responsiveness for very large data sets even in the presence of frequent disk failures. In Figure 4-10, we plot the MTTDL for a 20 Petabyte file system given various choices for the size of error correcting codes as well as various failure probability distributions. As can be seen in the Figure, depending on the various assumptions used, it is possible to curate this amount of data over many years depending on the strength of the error correction used. Figure 4-11 shows the data losses per year for a larger data set of 100 Petabytes using the ideas of declustered RAID and multiple data distribution so that several disk failures can be dealt with. Using these approaches it is possible to bring down data losses to very low levels. It should be emphasized that the failure models used here do not take into account truly catastrophic events that may bring down some portion

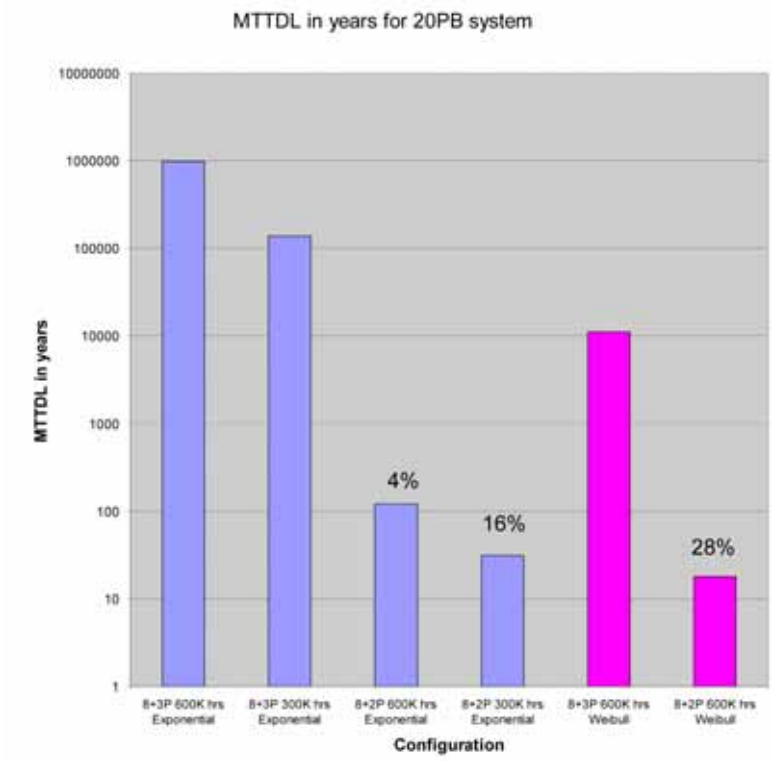

Figure 4-10: A plot of the mean time to data loss (MTTDL) for a 20 Petabyte data set under various assumptions of failure distributions and strength of error correcting codes [13].

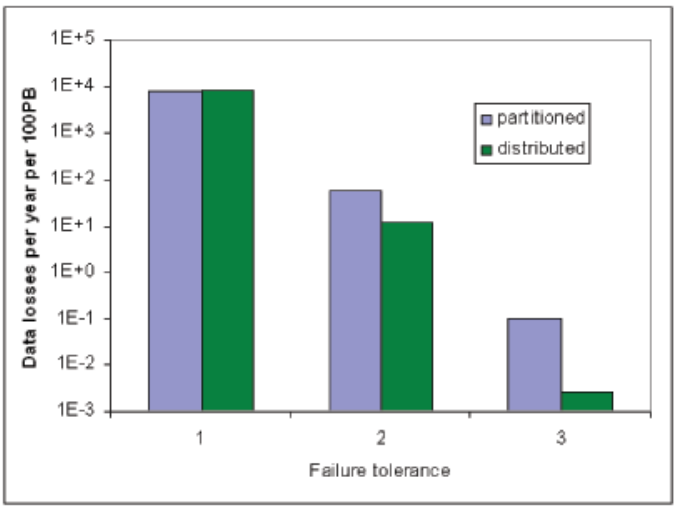

Figure 4-11: A plot of the data loss per year per 100 Petabytes comparing partitioned vs. declustered RAID as a function of the fault tolerance of the system [13].

of a facility. Barring such events it is possible with significant investment in software to deal with the data requirements of the DOD and other enterprises for some time into the future. We have said nothing however about the need to query this data. This will be discussed in the next Section.

5 HANDLING DATA IN DIFFERENT WAYS

Traditional DOD systems have typically been engineered as "turn-key" systems. The sensors, storage and analysis systems are often tightly connected. There is of course an advantage here: the turn-key system solves an immediate problem and appears cost effective for the particular problem at hand. However, as new sensors become available, a natural goal is to perform "multiple source" analyses by federating and fusing the data sources from the various sensors. This is made difficult by the use of turn-key systems where data acquisition methods and formats were not designed originally with the goal of future data fusion. As a result, turn-key systems cannot generally be gracefully evolved and the complexity of the data infrastructure increases. This is not a hardware issue; it is a software design issue. In this section we discuss some of the strategies utilized by large data providers such as Yahoo! and Google and examine some of the infrastructure and algorithms involved. This differs significantly from the use of turn-key systems. Instead, the objective is providing an infrastructure that enables generic investigation of the data. For various DOD applications, such an approach may offer significant benefit as we discuss below.

The approaches to handling large data in ways that are more architecture and system neutral and which support fusion of data will differ depending on the requirements for the timeliness of the information. We can broadly distinguish three cases:

Long time scale Here there is no critical timeliness requirement and one may want to establish results on a time scale of perhaps days. Applications which match well include retrospective analysis of multiple data sources, fusing of new data to update existing models such as geographic information systems or to establish correlations among events recorded through different information gathering modalities. This type

of data analysis lends itself well to a production or "batch" environment.

Medium time scale Such a time scale corresponds to activities like online analysis with well structured data. Typically this is accomplished in an interactive way using a client-server or "pull based" approach. We argue that this matches well to present day Service Oriented Architecture (SOA).

Rapid time scale In this scenario, one wants to be cued immediately for the occurrence of critical events. The time scale here may be very near real time. We will argue that a "push based" or event driven architecture is appropriate here.

We discuss ways in which data can be handled as guided by the timeliness requirements for information in the sections below.

5.1 Approaches to Long Time Scale Analytics

Many organizations have a variety of big data problems. Even if most effort goes into production processing, there is a need to accommodate incremental improvements and upgrades.

The experience of big Internet companies gives a recent approach for managing big data. Yahoo! and Google provide applications with homogenous infrastructure and a computing model implemented in software that together cover a large range of their big data processing problems.

A more conventional approach is to acquire hardware and software carefully tuned to the problem at hand. This seems to optimize initial costs, or floor space, or power consumption, or whatever shows up in the spreadsheet. The problem has been that it is very difficult to estimate the value of being

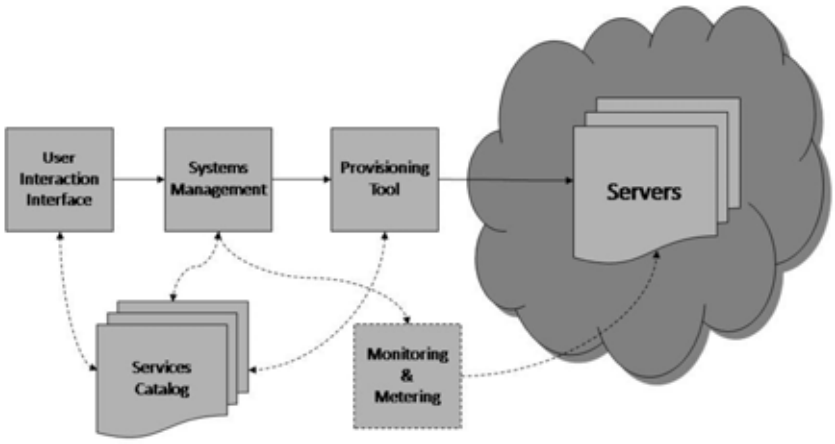

Figure 5-1: A diagrammatic representation of "cloud computing":. In this approach, the user sees a cloud of computers with a set of offered services. Provisioning is then performed based on the user's needs.

able to cope with an uncertain future. If an organization has to deal with different problems (and over time all organizations do), acquiring systems optimized for each problem adds complexity.

In the experience of companies like Yahoo, Microsoft, and Google it is better to build large data centers of essentially identical servers running essentially the same operating system, and require all applications to make use of these servers. Commodity servers are economical, and the common infrastructure is flexible and can be reallocated among applications fairly easily, especially as the machines can be loaded with essentially the same environment.

This is the basis of what is today known as "cloud computing". In this approach to deploying hardware, users view a homogeneous infrastructure (the location of which is made largely irrelevant by using high speed networking and virtualization). The way in which one interacts with the data is also quite different in that one delivers algorithms to data rather than use a data base to precompute various indices ahead of time. This approach is shown illustratively in Figure 5-1.

5.2 The Map-Reduce Archetype

Yahoo! and Google process large amounts of data using a map-reduce framework. Yahoo! has released an open-source implementation of this named Hadoop which also provides a file system architecture discussed below. Map-reduce works on data set sizes of Terabytes and up. Typically the data would be spread across multiple servers. A map-reduce job consists of a controller, M mappers, and R reducers. The controller might try to put the M mappers on machines close to the data. It breaks up the data into pieces and gives each mapper pointers to a set of hunks of data to process. Each mapper converts input records into output key-value pairs. The key-value pairs are sorted by key into R segments, and each segment is sent to a reducer, who sees its input in key order, and (presumably) processes it and writes a hunk of output. Many large computations can be organized this way, or by a sequence of map-reduces. One measure of success is that the current "terasort" (sort a terabyte of data from disk to disk) record was set by Yahoo! using Hadoop. For this application the mappers and reducers implement the identity map, just copying their data from input to output.

For a different example, one might have many documents, and want to index them by language and uniform resource identifier (URI). The mappers read the document and put out language as the key and URI as the value. The reducers create the index (possibly removing duplicates) directly, and could also count the number of documents in each language. The idea is shown graphically in Figure 5-2.

A very important aspect of this approach is that it is essentially embarrassingly parallel and is therefore ideal for parallel architectures. Because each map and reduce operation can be dealt with autonomously, one can envision the process as simply a set of tasks that must be accomplished but which are not dependent on one another. This makes it possible to use even

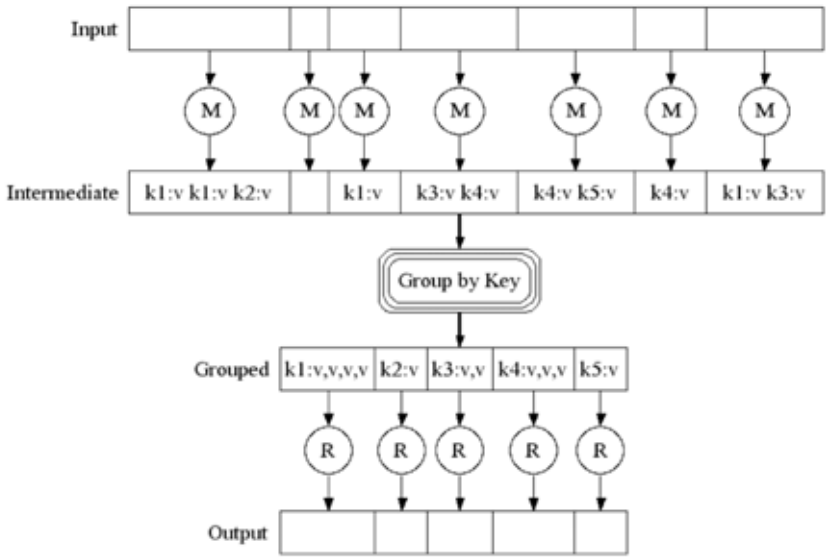

Figure 5-2: A diagrammatic representation of the map-reduce archetype. The mapper processes M emit keyword value pairs which are then grouped by key. This is then fed in key value order into reducer processes R which then perform some reduction operation associated with the keys and their values.

commodity hardware where hardware components will typically have a lower mean time to failure then enterprise class hardware. This makes it possible to simply monitor jobs and then just restart those that fail and in the process migrate them to other processors. This is particularly important if we are to contemplate Petabyte data sets and tens or hundreds of thousands of processing elements. At that scale the probability of node failure is significant as discussed in Section 4, and so one desires an approach that routes gracefully around such failures. The parallelism idea is also easy expressed graphically and is seen in Figure 5-3. Note there can be dependencies between various mapper tasks and associated reduction tasks. The reducer will wait until the appropriate answer is delivered. If a monitoring program sees that this has not occurred it can simply replicate the mapping process and provide it with the address of the target reduction.

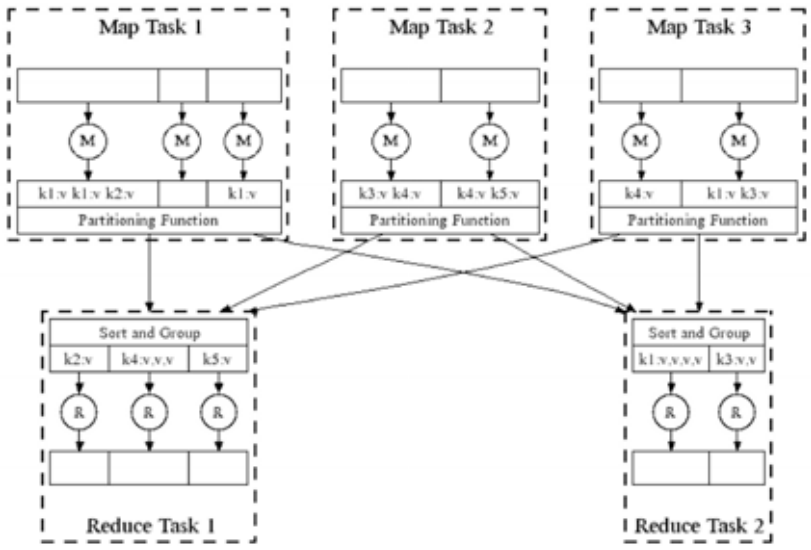

Figure 5-3: Parallel implementation of the map-reduce archetype

The canonical example of the use of map-reduce is counting the number of occurrences of each word in a very large collection of documents. The user would write the following pseudocode

```
map (String key, String value):
// key: document name
// value: document contents
for each word w in value:
EmitIntermediate(w,''1'');

reduce(String key, Iterator values):
//key: a word
// values: a list of counts
int result  = 0;
for each v in values:
result += ParseInt(v);
Emit(AsString(result);
```

The map function emits each word encountered along with a count (in this case just 1 indicating the word was encountered). The reduce function then sums the counts for a particular word.

There are surprisingly many types of data analyses which can be performed using the map-reduce archetype which is why we term it an "archetype". It represents a specific type of computational pattern which can be reused. We list some examples below:

Distributed grep The map function emits a line if it encounters a particular pattern. In this case the reducer simply emits the line to output. This allows one to search in parallel for strings matching some given pattern.

Count of URL access frequency In this case the mapper takes logs from web page requests and emits a 1 for each URL encountered. The reducer in this case adds the values for a given URL.

Reverse web link graph Here the mapper outputs <target, source> pairs for each link to a given target web URL found in a web page named source. The reduce function then concatenates the list of all source URL's associated with a given target URL and emits a pair <target, list (source)>. This is a key step for example in page ranking algorithms which are used in modern search engines.

5.3 The Hadoop File System

The above discussion illustrates alternative ways to query large amounts of data even over distributed data stores. However, in order to access the data one must be able to count on the reliability of the access mechanisms. As mentioned in the previous section the high performance community is developing solutions that use redundancy, caching and error correcting code in order to ensure that large data sets can be reliably curated.

If the goal is to query data repeatedly in order to extract various correlations, there are open source solutions available now that work on commodity

hardware. One such is the Hadoop file system. Hadoop is a distributed file system with many similarities to existing file systems but also has some important differences. It is designed to work on commodity hardware which typically has a higher failure rate than enterprise storage platforms. As will be seen below, it targets large data sets where the typical mode of access is "write once, read many times". It is not a good choice for data that changes frequently. The design criteria are as follows:

Hardware failure is likely The system is built on the assumption that hardware will fail. A typical Hadoop file system (HDFS) deployment may consist of thousands of servers each with some commodity storage. The assumption in HDFS operation is that for one reason or another some server or servers is always nonfunctional. The system is built to detect faults and recover gracefully.

Streaming data The idea behind Hadoop is to support archetypes like map-reduce where data is essentially streamed through the mapper and reduction processes. Map-reduce jobs are typically run in a batch mode. Hadoop is not an appropriate choice for applications that need random access to files. The emphasis here is throughput and not low latency.

Large data is the norm Hadoop was designed for data sets that typically will not or cannot be stored on one central storage system. Data sets are typically Gigabytes or Terabytes in size. Scalability to these sizes as well as the number of storage nodes required to store such data is a key goal of the file system.

Coherency As stated previously, Hadoop is most appropriate for a "write once read many" access model for files. The idea is that once the file is created it will not change. This is indeed the model one would use in surveillance although products derived from these files and files from later surveillance activities would change but this information would not need to be stored using Hadoop. This approach simplifies issues

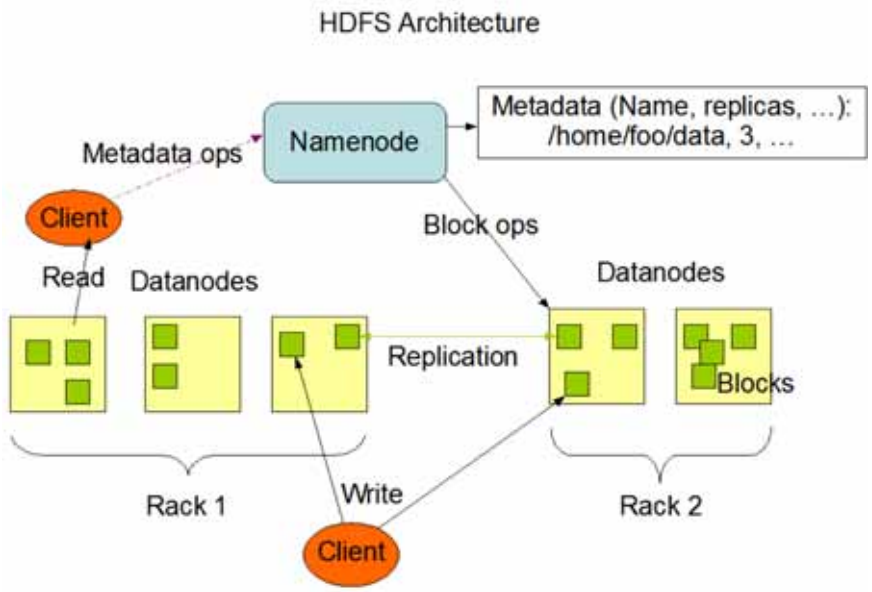

Figure 5-4: Architecture diagram for the Hadoop file system

of data coherency. Many retrospective analytics applications like web crawling fit these assumptions.

It is cheaper to move computation than data The overall approach to analysis using Hadoop is to move a computation close to the data rather than move the data to a central point where computation takes place. This minimizes congestion and is more scalable in that there are fewer load imbalance bottlenecks due to data motion or computation. Hadoop provides interfaces to adjust the "affinity" of a computation for a particular location where important data resides.

Architecture neutrality The design of Hadoop makes no assumptions about hardware capabilities and so it is ideal for analysis using heterogeneous architectures and storage systems.

HDFS uses a master-slave architecture which is shown in Figure 5-4. An HDFS cluster consists of a single master server that manages the file system name space and regulates access to files by clients. There are also data nodes (typically one per node in a cluster) which manage the storage of

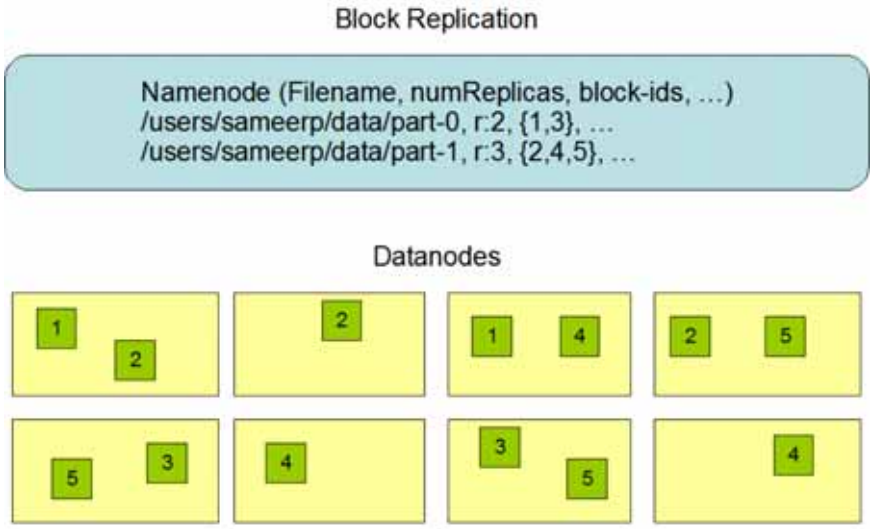

Figure 5-5: File segment replication strategy for the Hadoop file system

files. The overall approach is to simply let users store flat files rather than impose any type of database structure. We will comment more about this below. Internally, files are split into blocks and blocks are then stored on data nodes. The mapping of the files is the responsibility of the head name node. The data nodes then serve read and write requests from clients of the file system including tasks like opening and closing files etc. HDFS supports a typical hierarchical file system organization and from the point of view of the client it is possible to perform most of the usual file operations like opening, closing, moving, and renaming. Importantly the file system does not support editing in a file although it can support appending data.

The key property that leads to the reliability of the file system is the aggressive use of block replication. Each file is stored as a sequence of blocks and these are then replicated for fault tolerance. The replication factor and file block size are all configurable for a given application. As stated above, the files are "write once" and only one client at a time can write. The system queries the storage nodes periodically and receives a "heartbeat" that implies the node is functioning and also a report of which blocks are on which node. The placement of the replicas is also carefully designed so as to be "rack

aware" so that if a node goes down the system goes to a rack close by to access the data. Typically, a replica is also placed on a distant node so that in case of some more catastrophic failure the data can still be retrieved although access will be slower. The replica placement process is an important optimization problem for which further research is required. The replication idea is shown graphically in Figure 5-5.

5.4 Databases in the Context of Large Data Sets

The scientific community faces the challenge of storing, searching and accessing large data sets, as well as the need to archive data in a robust and enduring way. Databases would appear to offer just what is needed to accomplish this, as in some ways the problem appears superficially similar to that faced by financial institutions, where databases have long been used. In fact database design and implementation has largely been driven by the needs of financial institutions, not science. Databases have therefore evolved to deal well with supporting concurrent transactions, dealing with both numerical and text information.

Broadly speaking the segment of the scientific community that is pushing the forefront of large-data science has been disappointed with the capability and the performance of existing databases. Most projects have either resorted to partitioned smaller databases, or to a hybrid scheme where metadata are stored in the database, along with pointers to the data files. In this hybrid scheme the actual data are not stored in the database, and SQL queries are run on either the metadata or on some aggregated statistical quantities.

Partitioning the database into disjoint smaller pieces is always an option, of course. For instances where there is no requirement to have a global perspective this can work well. On the other hand if one wanted to run a

conditional query that was not confined to one subspace then the partitioning can extract a significant performance penalty, as the database indices do not span the distinct partitions.

In the sections below we detail some of the database shortcomings that the scientific community has encountered, and we end with a look ahead.

5.4.1 Dealing with uncertainties

As distinct from accounting information, scientific data have uncertainties associated with nearly every measured quantity. In high-dimensionality parameter space, one would like the ability to efficiently run sophisticated queries that take into account not only uncertainties but also the covariance between quantities. The data types in current databases do not easily lend themselves to rapid and efficient interaction with the stored data, in the context of underlying uncertainties.

5.4.2 The data provenance problem

The data reduction process starts with "raw" sensor data of some type, and these data are then typically run through a sequence of processing stages, each of which has source code and parameter settings that change over the course of the experiment. There is no universally accepted way to store (in a fashion that allows straightforward reproduction of results) not only the reduced data, but also the code version and parameter choices that produced the data. One could imagine a hybrid of database technology with version and configuration management software, at the middle-ware level. We are unaware of any robust, open source, platform-independent solution to this problem.

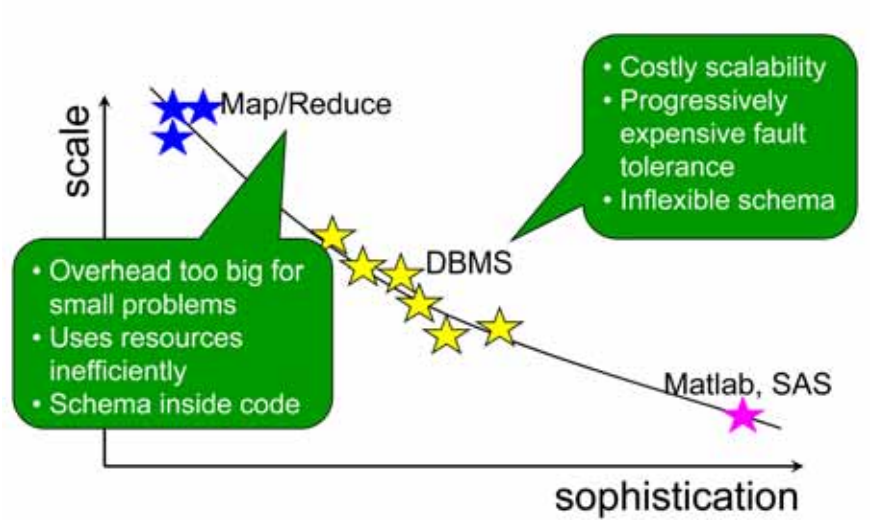

Figure 5-6: A graphical comparison of data mining approaches [3].

5.4.3 Some disappointing experiences to date with existing databases

The astronomy and high energy physics communities shared with us their experiences to date with using database technology to support 100 Terabyte scale data sets. The experience of the BaBar project was illustrative of the broad disappointment we heard. This group started with using a commercial database product, but problems with both performance and licensing costs drove them to drop this in favor of an open source partial solution. They store their metadata in a database, but the actual scientific data are kept in a file system that meshes with the database. Some groups have described the time consuming process of needing to generate new index information whenever new data are ingested into the database, and appears in some cases to scale poorly as the size of the data set grows.

5.4.4 Evolution of databases

The difficulties described above with conventional database systems are also present for DOD/IC applications. Data intensive enterprises such as Google and Yahoo! do not use strict database approaches in their work and

instead use a more homogeneous approach to data analysis which makes use of the map-reduce archetype. The tension between conventional data base technology and approaches such as those embodied in technologies like map-reduce and Hadoop can be seen graphically in Figure 5-6. One can view the tradeoff as one of sophistication vs. scalability. At the high end of sophistication are analytical approaches like Matlab, Excel, Access, etc which provide ease of use and a rich set of analysis tools. But at present, these are designed for the workstation market and it is not possible to apply them to large data (although there is ongoing work in this direction). More scalable, but less user friendly, are data base management systems which have traditionally been applied in this arena. The advantages are that these are very efficient and highly tuned but scalability is very expensive and as has been discussed above, the adaptation of schemas has proven difficult. Archetypes like Map-reduce running on infrastructure like Hadoop are very scalable but they come with a high overhead and cannot be used for smaller problems. Here one embeds the schema in the mapper and reduction functions so that while the system is very flexible it requires familiarity with algorithmic programming.

Interestingly, this has created pressures to improve data base flexibility at one end while providing more capable interfaces for approaches like Map-Reduce. Database manufacturers are redesigning their engines for multiprocessor scalability and also examining the use of less traditional schema. An example is the Vertica effort of Stonebraker and his colleagues which attempts to address the issues that have been raised by the scientific community. There is also significant ongoing work in endowing the highly scalable Map-Reduce approach with better interfaces such as for example SQL. For example, Widom has researched the requirements to create streaming data base architectures that extend the familiar SQL programming model to data streams. There are implementations of this idea for example in the Hive language which is used by Facebook for their data warehousing. As a result, we can expect continued improvement in this area with a concomitant benefit for DOD/IC data intensive applications.

5.5 Probabilistic Streaming Algorithms

As can be seen from the previous sections, storing and curating large volumes of data is feasible. However, querying the data can be prohibitively expensive particularly if we require exact answers to our queries. Recall that we forsee the querying of possible hundreds of Petabytes. Even with an efficient map-reduce approach the time to develop exact responses to queries may be prohibitive. Scanning all the data is effectively at least a linear time operation so as to ensure all possibilities are exhausted. Large data users such as Amazon face such challenges when they engage in customer analytics. For example, the familiar quote "People who bought this product also bought this..." exhorting one to purchase additional items is an example of "collaborative filtering". A similar requirement emerges when one wants to check if a given document matches a library of stored documents or if a fingerprint possesses a match in a fingerprint database. In this case even linear time algorithms might be prohibitive.

One approach is to use probabilistic algorithms which are amenable to streaming data. We give some examples below. The main point of this discussion is not to provide specific solutions, but to illustrate the power of such algorithms and to recommend that the computer science community be fully engaged in developing such algorithms for DOD/IC applications.

5.5.1 Bloom filters

A Bloom filter is a very simple probabilistic technique for representing a set in a very memory efficient way so as to facilitate membership queries. Bloom filters were developed in the 1970's as a tool for database systems. Recently, their use has been proposed for networking applications which makes them important for the type of distributed analysis required for data fu-

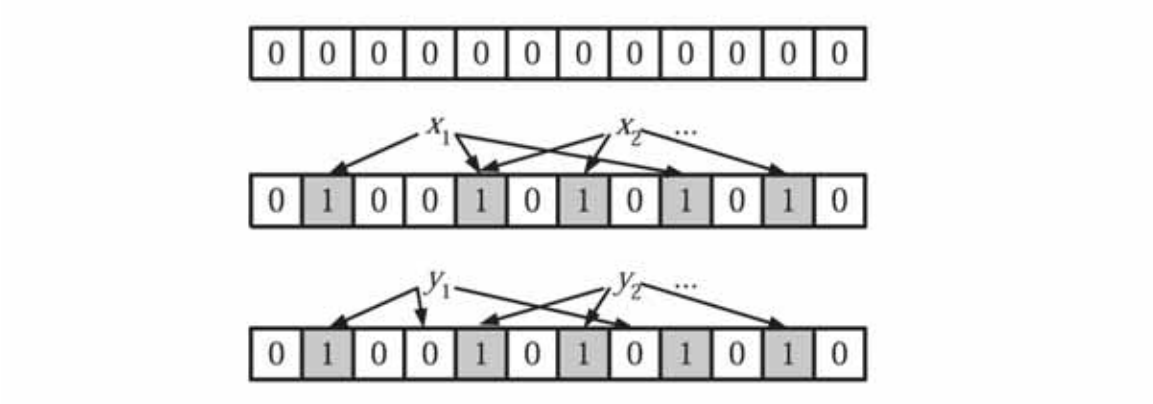

Figure 5-7: An example of a Bloom filter [17].

sion. For example, Bloom filters can be used to summarize content to aid collaboration in peer-to-peer networks.

The Bloom filter principle is that whenever a list or set has to be interrogated and space is at a premium, then a Bloom filter is useful as long as the effect of false positives can be mitigated. A Bloom filter for representing a set $S = \{x_1, x_2, \ldots, x_n\}$ of n elements is described by an array of m bits, initially all set to 0. A Bloom filter uses k independent hash functions labeled $h_1, \ldots, h_k$ with a range $\{1 \ldots, m\}$. We assume the hash functions map each item in the set to a random number which is uniform over the range of hash values $\{1, \ldots, m\}$. For each element $x \in S$, the bits $h_i(x)$ are set to 1 for $1 \leq i \leq k$. A given location can be set to 1 multiple times. To check if some item y is a member of the set S we hash the value y k times and then check whether all $h_i(y)$ are set to 1. If they are not, then the item y is not a member of the set S. If all $h_i(y)$ are 1 then we assume $y \in S$ but there is a probability of a false positive. This is shown graphically in Figure 5-7. It is important to note that in using this idea we assume $kn < m$. A false negative is impossible using this approach. False positives may be acceptable provided their probability is small [17].

The probability of a false positive can be estimated provided the hash functions are perfectly random. After all the elements of S are hashed into

the Bloom filter, the probability that a specific bit is still 0 is

$$p' = \left(1 - \frac{1}{m}\right)^{kn} \approx \exp(-kn/m). \tag{5-1}$$

If we let ρ be the proportion of 0 bits after all the n elements are inserted in the table. The expected value for ρ is p'. Conditioned on ρ, the probability of a false positive is

$$(1 - \rho)^k \approx (1 - p')^k. \tag{5-2}$$

It turns out because of the concentration of the distribution of ρ about its mean which we do not discuss here, this is in fact the false positive probability. Provided one can cope with the false positive rate, such a probabilistic approach is of value [17].

5.5.2 Minhashing and locality sensitive hashing

We were also briefed by Prof. Jeffrey Ullman of Stanford University on several algorithms that work well for large scale data mining where the data cannot be exhaustively queried. These algorithms are used for what is called similarity search. The objective is to find pairs of objects that are similar in a collection of a set of objects. Similarity can be defined in many ways, and will depend on the application, but a particularly useful definition is that of Jaccard similarity which is the size of the intersection of two sets divided by the cardinality of the union of the two sets. Among others, important applications of Jaccard similarity include collaborative filtering discussed above and document similarity. Here documents are represented by their sets of what are called k-shingles which are strings of k consecutive characters. Other applications include fingerprint checking after a finger print is discretized in terms of its minutiae, and entity resolution where one wants to consider similarity of attribute/value pairs in order to match descriptions of individuals[26].

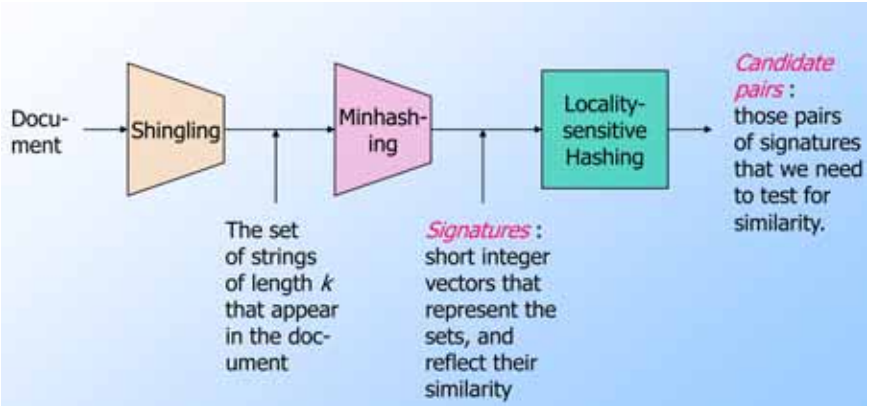

Figure 5-8: A schematic for the use of minhashing and locality sensitive hashing in determining similarity of documents [26].

The ideas of minhashing and locality sensitive hashing share the philosophy described above in the use Bloom filters. The application of minhashing requires the construction of small signatures for sets so that the Jaccard similarity of two sets can be determined from the signatures rather the examination of a detailed comparison of each record. Locality sensitive hashing uses this idea to focus on pairs of sets that are likely similar without looking at all the pairs. The flow of operations for determination of document similarity is shown in Figure 5-8. These types of algorithms are very useful when the sets are so large that checking all pairs for a match takes too much time. Instead the idea is to determine some candidates and then examine these in detail.

Consider the sets to be compared as represented by a matrix of zeros and ones. Let the rows label the individual objects and the columns label the various sets of objects. We place a one in a given row and column if a particular object appears in a given set. To compute the similarity of two sets of columns we count the rows where a given object appears in both columns and then divide by the number of rows where at least one object appears.

Clearly there are four types of rows as described by the table below:

	C_1	C_2
a	1	1
b	1	0
c	0	1
d	0	0

If we designate A as the number of rows of type a, B as the number of rows of type b etc., then the similarity of the two columns C_1 and C_2 or $Sim(C_1, C_2)$ is given by

$$Sim(C_1, C_2) = \frac{A}{A + B + C}.$$

The idea of minhashing is to imagine permuting the rows of the matrix randomly. We then define a hash function $h(C)$ which is the number of the first row in which column C has a one (in the permuted row order). We then use an independent collection of such hash functions to create a signature with say 100 such hash values. It turns out the probability over all permutations that $h(C_1) = h(C_2)$ is the same $Sim(C_1, C_2)$. It is not hard to see that the similarity of two signatures is the fraction of the rows over which they agree. From a practical perspective we would not want to permute these rows physically but a good approximation to this is to pick say 100 hash functions. For each column C and each hash function h_i, create an array $M(i, c)$ for that minhash value. We then take the minimum value of the hashed value that corresponds to to non-zero values of $M(i, c)$. [26]

This allows us to replace whole sets (which are columns of our matrix) by short lists of integers. But to compute if something is similar to something else we need to complete all pairs so we still have a problem in terms of the number of operations to be done. Instead we map signatures to buckets of signatures with the objective that two similar signatures will end up in the same bucket with high probability. If two signatures are not similar there is high probability that they don't appear in a given bucket. Now we consider the signature for each column (which recall is a set) as a column of what we call the signature matrix S. We then divide the rows of S into b bands of r rows each. For each band, we hash its portion of each column to a hash

table with many buckets. Our candidate column pairs are those that hash to the same bucket for $\geq$1 band. The values of b and r must be tuned to catch most similar pairs and avoid the dissimilar pairs. A counting argument shows that by using multiple bands it is possible to get high probability that similar objects appear in similar buckets. The ideas are similar in character to those presented above in the discussion of Bloom filters [26].

Algorithms such as these can be very useful when dealing with large data sets. It will become essential to use such ideas as the DOD/IC face analyses of Petabyte data sets. Continued research and development in this area therefore would be of benefit.

5.6 Service Oriented Architecture

In this section we consider those approaches for data analysis that are appropriate for more intermediate time activities. Here we endorse the use of service oriented architecture (SOA) that is currently being explored in a variety of DOD and IC research efforts. An SOA is an architecture that relies on service orientation as its fundamental design principle and whose chief characteristics are modularity and the ability to access the service remotely. The idea is to create large scale components that perform a range of related tasks and provide an interface for these components so that these functions can be accessed via remote procedural calls by providing a well documented "service" typically over the web.

Services are meant to be largely autonomous units of functionality and communicate with each other via a protocol that typically implements a remote procedure call (RPC). In many ways they are similar to client/server architectures that use RPC or a request/reply communications approach. In contrast to C++ classes the atomic level objects of an SOA are generally substantial programs in their own right. The calling hierarchy associated with

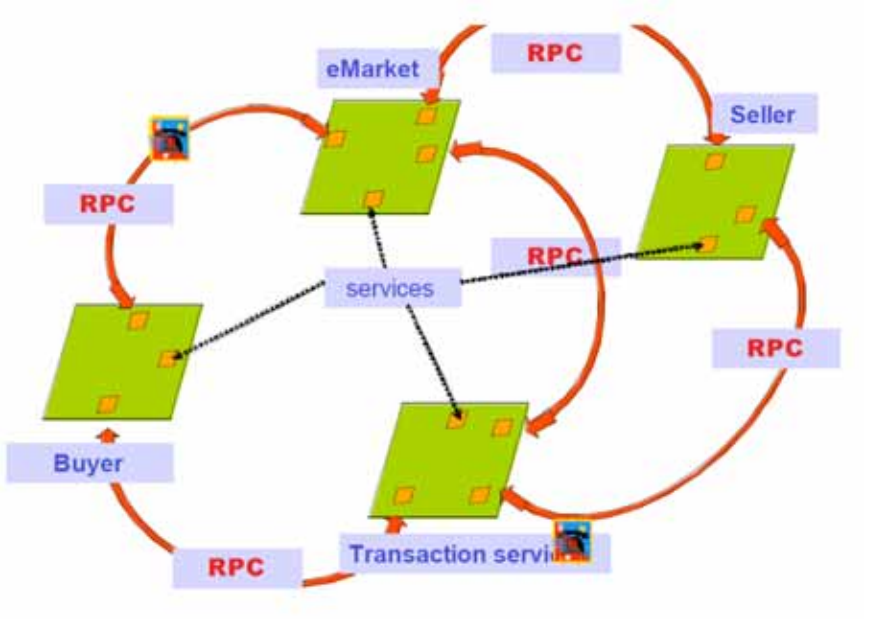

Figure 5-9: A graphical representation of an SOA for market based transactions [16]

a typical application is not as deep as one sees in typical C++ class diagrams. Applications are typically designed using special software which can discover the services over the web and then orchestrate the flow of information.

A simple example of such an application might be the purchase of services through a market application. This is shown graphically in Figure 5-9. A buyer of services would make calls to a marketing service which would link in a selling service and some sort of support for transactions. SOA is attractive for DOD applications where large data stores need to interoperate and where fusion of their data is required at a higher level. We were briefed on several projects where different agencies were proposing making their products available so that larger applications could be constructed that addressed more specialized needs [2]. The DOD and IC have already built and utilized several SOA applications. We were briefed on a system developed by NRL called EVIS which provides weather information (displayed in Figure 5-10). The important advantage in doing this is that the results can then be directly embedded so that DOD decision makers can fuse several data sources

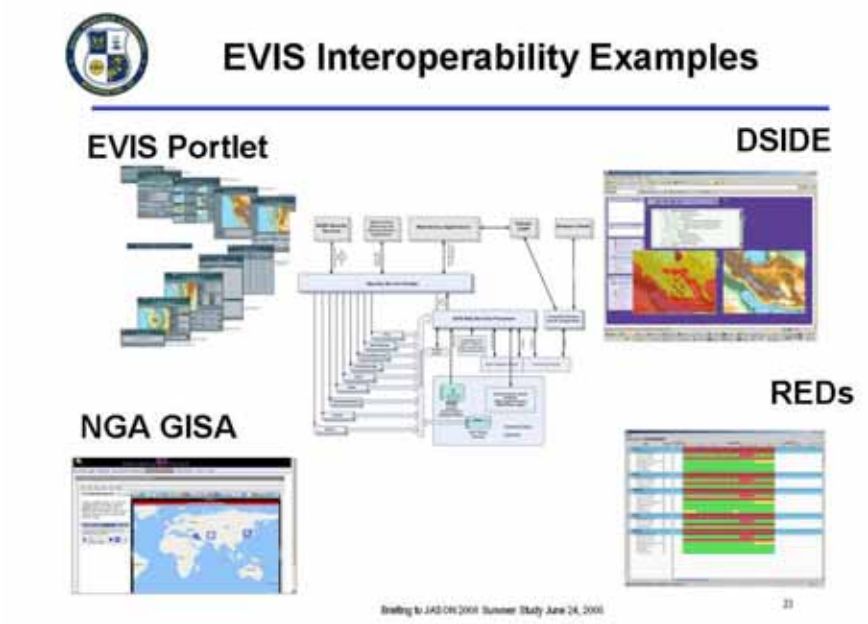

Figure 5-10: An example of a service oriented architecture - the Environmental Visualization (EVIS) can interoperate with information systems from several DOD/IC agencies [2].

in a straightforward manner. One potential issue with this approach is the problem of scalability and latency. Because of the use of the request/respond approach it is possible that requests will not complete in a timely way as they are waiting on other requests. This can be avoided through careful attention to design and ensuring that time critical services respond on an appropriate time scale.

We were also briefed on the use of SOA to integrate a number of DOD/IC services in a project called Blackbook 2 developed by Johns Hopkins University and funded by IARPA. Blackbook is built as a data integration framework and provides a common data representation format using semantic web technologies as well as provision for web services. Rather than integrate vertically Blackbook takes very much the point of view of this study in establishing a data sources layer where data (not necessarily data bases) resides. Eventually the data can be queried using ideas such as map-reduce with the results integrated though an infrastructure layer which provides a

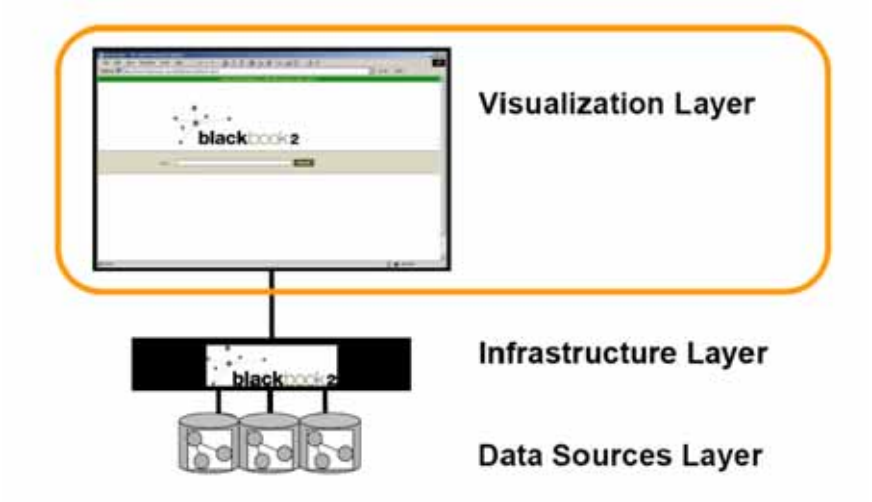

Figure 5-11: The architecture of Blackbook2. Data sources are integrated via an infrastructure layer. Results can then be visualized in a number of ways via the visualization layer [6].

variety of services. The use of web technologies such as Resource Description Framework (RDF) and XML makes it possible to translate from a number of data sources which can include relational data bases all the way to unstructured text. The user interface provides a number of ways to visualize the information including a Google like search, spread sheets or even geospatial information systems if the data being analyzed is geographic in nature. The architectural diagram is shown in Figure 5-11. An example of the graphical user interface is shown in Figure 5-12 [6].

Blackbook represents an important effort to remove the "stovepipes" associated with some traditional DOD/IC data collection enterprises. The use of modern technologies including architecture neutral data storage and the development of an infrastructure that can be extended in numerous ways is an important step forward. Because the project is geared towards information sharing among the DOD and IC agencies there is a security model already built into the transactions. The use of data neutral approaches such as Map-Reduce and reliable file systems such as Hadoop will be of benefit in dealing with the emerging DOD/IC large data requirements.

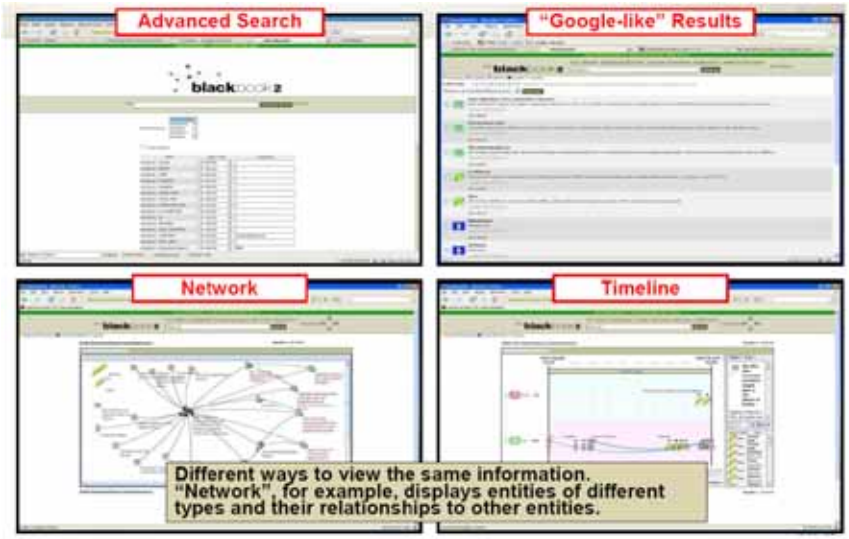

Figure 5-12: An example of the multiple data representations available within Blackbook 2 [6].

5.7 Event Driven Architecture

While SOA is of great use in aiding the data fusion and integration problem, it is not directly useful for analyzing events on a rapid time scale. For such situations, an event driven architecture may be more appropriate. An event driven architecture (EDA) is an approach to software design that deals directly with the production, detection, analysis of and reaction to, various events. An event is simply a significant change of state associated with some data that is being constantly monitored.

Programming for EDA applications differs from that of SOA although one can merge one with the other. EDA applications use a publish/subscribe model where loosely coupled software components subscribe to event streams and then either react by actuating some response or by emitting subsequent events to other components. The key idea behind this approach is asynchronous broadcasting or "push" of events. The events may trigger subsequent action but do not themselves define actions. An example of an

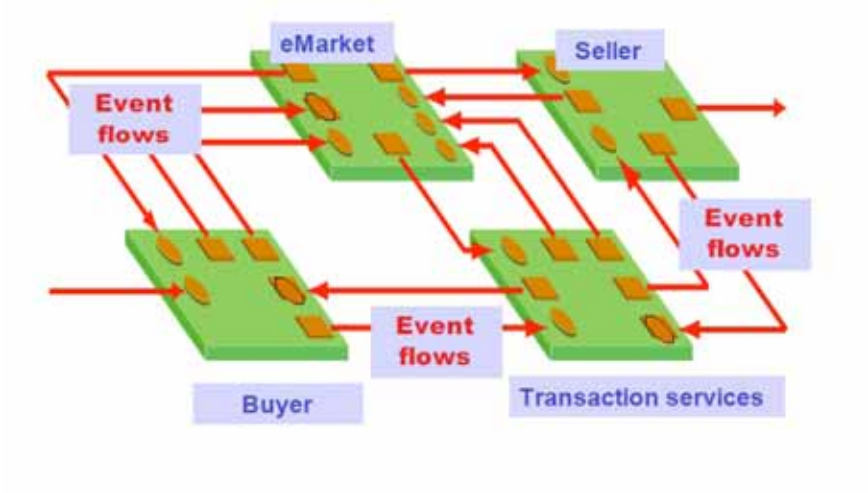

Figure 5-13: An example of an event driven approach to market based purchases [16].

event driven approach to a system which executes market based purchases is shown in Figure 5-13. All components emit event streams to which other components subscribe. When relevant events arrive to various components analysis then triggers actions or results in further generation of events. An important advantage over traditional SOA approaches is that event driven systems are more responsive and are by design more tuned to dealing with unexpected events.

The structure or content of an event depends on the application. Typically, it consists of two parts: a header which labels the type of event and other important metadata such as time and geospatial information, and the event body which would contain important information (image, text etc.) that must be analyzed in order to develop a response if the event triggers subsequent analysis and response. An event based architecture is generally composed of four layers:

Event generation The event generator senses some occurrence and generates the information comprising an event. It can be anything from a sensor that triggers on some change in a scene or even an e-mail

client that receives a certain type of message. The event generator will synthesize the appropriate data for handling down stream. The structure of the event will depend strongly on the application and there is benefit to having analysts in the loop when the overall design of the EDA is contemplated. The event data will most likely be transformed downstream so standardization at this stage is not crucial. However, for DOD/IC applications that rely on spatio-temporal events obvious components of valuable event data are the time and GPS coordinates associated with a given event. Interestingly, financial firms now use GPS satellites with their highly accurate clocks to provide a standard time and location for the generation of financial events.

Event data channel This is simply the mechanism whereby the event data is transmitted to some event processing engine. It could be an IP socket or even something as simple as the generation of a file. In an event driven system the events will not appear ordered by the time they were generated (as in modern packet driven networks) simply because the latency associated with the channel and the high volume of events make this impossible. In addition, the philosophy of EDA is to process the events in near real time. This makes the use of time and spatial information very critical in DOD/IC applications.

Event processing Here the event is identified and actions are triggered. For financial applications this could be that the price of some commodity has reached a certain level triggering a transaction but it could also be a signal to focus further analysis on some location and to thus generate further events to cue other resources.

Downstream activity This describes the consequences of an event.

Event driven programming is already quite established. For simple event processing one simply actuates consequences based on simple occurrences. In a window-based GUI system a simple event might be a mouse click on a menu triggering the display of the menu or the activation of a temperature control

system. Event Stream Processing is the ongoing analysis of a stream of events where both ordinary and extraordinary events occur. This is the style of programming used in financial transactions and has spawned the design of stream based data management systems. Academic work in this area can be found for example in the design of the Continuous Query Language (CQL) developed by Jennifer Widom and her colleagues at Stanford. CQL is a streaming data base language that provides some of the analytical capabilities of SQL but for stream data types. This work has already evolved into the commercial sector with offerings such as Streambase.

The state of the art in this area is known as Complex Event Processing [16]. Here one examines patterns of seemingly ordinary events to determine that a larger more complex set of events has occurred. The set of events may occur over some period of time much longer than the natural frequency of ordinary events and, as a result, deeper pattern analysis is required. Correlations may be sought among temporal or spatial sets of events with the goal of detecting anomalies.

Traditionally, EDA is being applied in areas such as financial transaction information systems where it is necessary to deal with on the order of 10^5 events per minute with the use of complex event processing in order to detect larger shifts in a particular market. EDA is also routinely applied today in the design of controllers for DOD weapons systems where several systems such as radar, targeting and fire control must interact on a very short time scale. The field of EDA is quite well established, but does not seem to be employed in the type of enterprise-wide data analysis discussed here. It is seems not to be used extensively as a tool for data sharing among diverse DOD and IC organizations. Given its use in DOD contexts such as fire control systems and the inherent distributed and loosely coupled nature of this type of information flow it may be appropriate to explore this paradigm more throughly particularly where one must respond rapidly.

The event driven approach outlined above matches with the “activity

model" as briefed to JASON by Gordon Ainsworth. The idea is to focus on activities in various areas of interest and to correlate these with previous activities. The intelligence value of this approach was demonstrated recently in the capture of high value targets in Iraq. The actual analysis was a result of significant spade work by teams of analysts manually assembling various sources of intelligence. It would therefore be of interest to investigate whether EDA tools could make this type of analysis more efficient in the future.

5.8 Metadata Considerations - The Role of Registration

We close this section on various approaches to the handling of data with a short discussion of metadata requirements. As discussed above in Section 5.4.2, the data provenance problem remains an issue in understanding the pedigree of data after it has been processed from raw sensor data. One observation however, is that for image data the use of proper and accurate georegistration in tagging the data may be essential in facilitating the automatic correlation of image information in disparate databases and thus aiding in the analysis of complex events. Surprisingly, such registration is not yet standardized within the DOD and IC communities.

To be more precise, we define geolocation as the referencing of a particular location on the earth to a suitable absolute space-time earth-based coordinate system. Coregistration is defined as the referencing and cross registration of data in two or more data sources to a mutually reconciled coordinate system. While this is certainly important and useful for analysis, the ultimate goal should be georegistration which we define as the referencing of data or derived information of two or more data sources to some absolute coordinate system. It is the latter approach which will be of greatest value for fusion of disparate image data.

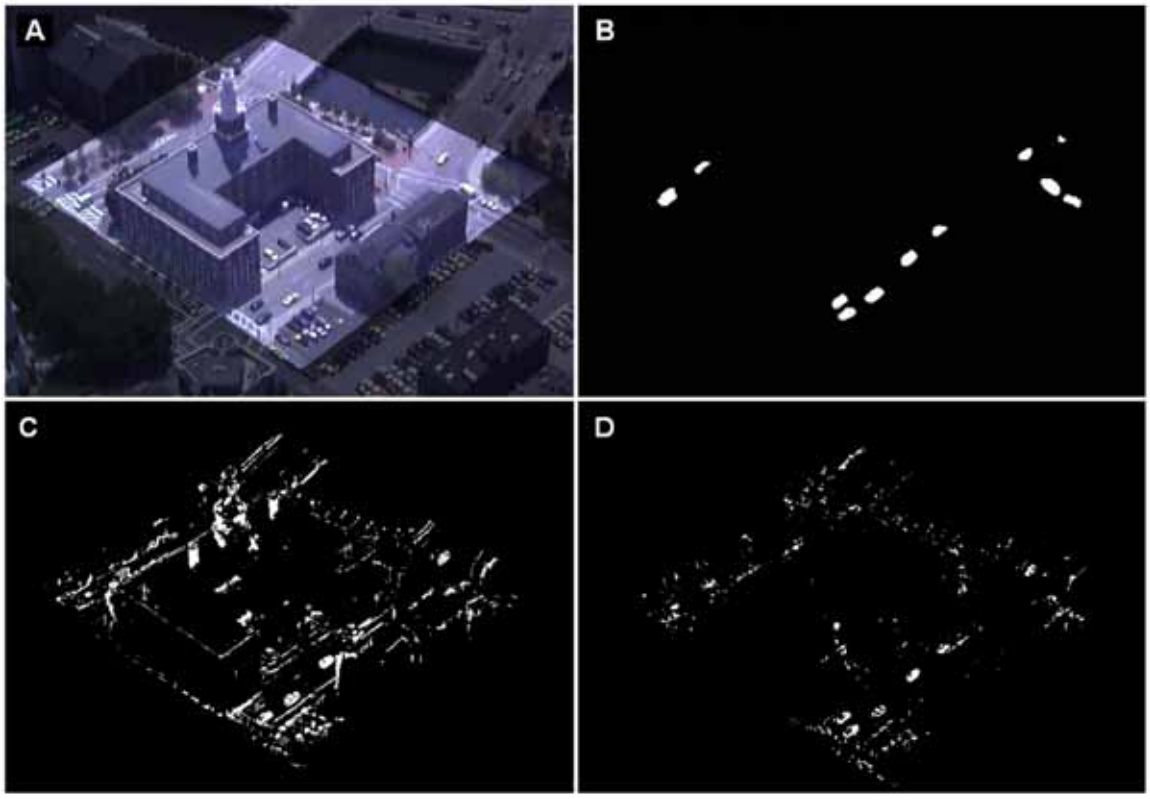

Figure 5-14: Voxel based approach to change detection of images Panel A shows an artificial but as yet unseen image. Panel B shows hand marked ground truth changes made to the image. Panel C shows the results of a planar change detection algorithm applied to the image. In Panel D, a probabilistic voxel-based algorithm is applied with improved results [20].

As more data is acquired from wide area surveillance at increasing resolution one might envision that the information could be used to update existing image repositories and then change-detection algorithms could be employed to cue further surveillance. There are still challenges associated with this approach because each surveillance platform must correct for its own attitude and parallax. Some interesting work on this problem has been published recently by Thomas Pollard and Joseph Mundy on "change detection in 3-D world". The authors use a 3-D voxel based approach to describe changing scene data acquired from a sequence of images that are taken by cameras with an arbitrary but known pose. By using a Bayesian approach, the authors develop a 3-D model of probability distributions which can be

continually updated as more imagery becomes available. These distributions can be used to infer whether change detection is the result of real changes or whether the change has been triggered because of occlusion of surfaces which are due to the fact that all images are 2-D projections of 3-D scenes. An example of this work is shown in Figure 5-14. Panel A shows an artificial but as yet unseen image. Panel B shows hand marked ground truth changes made to the image. Panel C shows the results of a planar change detection algorithm applied to the image. Note the appearance of fictitious change due to the attitude effects of the camera arising from the vertical but unchanged facets of the image. In Panel D, the probabilistic voxel-based algorithm is applied which still shows some fictitious change in addition to the actual change but the false positive rate is lower than that obtained from conventional 2-D based change detection algorithms. Moreover, continuing imagery and update of the voxel-based distribution functions have been shown to lead to improved results with ROC curves that show increasing true positive detection for a given false positive fraction. What is critical here however, is that the registration of the data must be the same for any given camera exposure. This is an important emerging area which when coupled with a consistent georegistration of image data would provide a useful basis for further development of change detection algorithms that could then cue analysts based on a number of image sources but with a hopefully acceptable false positive rate [20].

6 PROCESSING CLOSER TO THE SENSOR

Over time the capabilities of DOD sensors for wide area surveillance have grown impressively. This is best illustrated in Figure 6-1 which comes from the briefing of Dr. Mark Duchaineau from LLNL. If we assume that a pixel from a modern airborne sensor covers a square meter, then one can measure area coverage by counting pixels. In current practice, the data from a large sensor is collected and then stored using on-board storage on the airborne platform. After surveillance is complete, the data (in fact the disks themselves) are sent to a ground station for processing. Despite the latency of this approach, the impressive surveillance coverage afforded by present day sensors such as the Sonoma system has provided very valuable information. As can be seen from the Figure, the first 4 Mpixel Sonoma system fielded in 2003 was capable of imaging over 10^6 square meters, roughly the size of a city block with a repetition rate of 2 Hz. The Sonoma system fielded in 2004 imaged over 10^7 square meters, roughly the size of a small city. In 2009, MIT Lincoln Labs will field a system that can image 10^9 square meters, which is the size of a major city and it is projected that in 2010, the DARPA Argus system will be able to image an area that is almost the size of the Los Angeles basin at a repetition rate of 15Hz.

However, if one were to contemplate actually transmitting the information in real time as it is collected, the data rates required become prohibitive. For comparison the transfer rate of HDTV is shown in Figure 6-1. Even with present day sensors like Constant Hawk transfer rates of tens to hundreds of Gigabytes per second are required. Note this is not a storage issue but a bandwidth issue. Typical bandwidth available today is in the range of 100-200 MBits per second whereas the figure indicates something like 300 Gbits per second is required.

While the latency in delivering data from modern day sensors is acceptable for forensic analysis, it is not acceptable for time critical analyses. In this section we briefly discuss some strategies for processing sensor data closer to the sensor platform itself. This is not a substitute for storing all the data for later analysis but could be of use for time sensitive collection.

An obvious strategy is compression, but we were briefed that DOD makes significant use of modern compression technologies. Frame by frame compression using the JPEG 200 standard is well established as is the use of video key frames and the subsequent transfer of frame updates as is done in full motion video compression. However, the requirements will become more and more stringent. As is shown in Figure 6-1 a 250 fold increase in data rate is expected by 2010.

Overall a factor of 1500 is required to compress the information. Compression from JPEG will yield only a factor of 10; the use of key frames as in the H.264 standard yields a factor of 100 in compression but we are in need of over a factor 1500 in compression. An obvious strategy is to do on-board analysis and transmit the results in real time rather than the compressed image [7].

6.1 Use of GPUs for On-Board Processing

We were briefed by Mark Duchaineau of LLNL on the proposed use of a real time optimally adapting mesh so that analysts can selectively focus on objects of interest. The idea is to build an adaptive multi-resolution hierarchy of images directly from the raw sensor feed on-board the observation platform. Then depending on requirements, further analysis would be performed and sent to the analyst. For example, one might want to track some moving object in a scene. The goal then is to develop algorithms to isolate this motion and send only the object tracks (along with spatial infor-

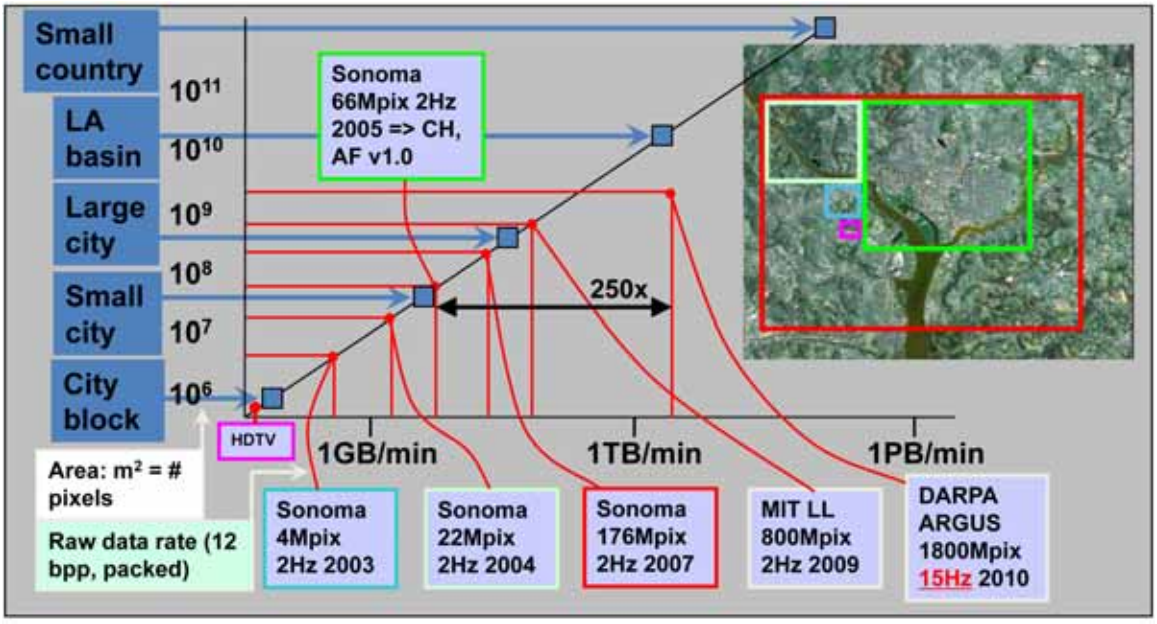

Figure 6-1: A plot of sensor capability vs. data rate for past, present day and future sensor systems [7]

mation) to the analyst. Because the platform is in motion one must register the imagery or it is impossible to follow any motion. But in addition, because of parallax effects there remain motion artifacts if one performs only registration. Duchaineau and his colleagues have developed "dense image correspondence" algorithms which utilize the overall evolution of the entire image to further stabilize images. This makes it possible to more easily identify real moving objects and it also provides for improved compression of the resulting scene. Figure 6-2 shows the results of this approach.

A remaining challenge is to implement the compression and extraction algorithms on hardware that would operate on the sensor platform. Graphics processing units (GPUs) show great promise for this as many of the operations needed are implemented optimally on the processor itself. In addition, current and future generations of GPUs are fully programmable. For the dense correspondence required to isolate movers one GPU is capable of processing at a rate of 22 MPixels per second. An array of 16 GPUs can support 176 MPixels at the required 2 Hz repetition rate. The only issue that has

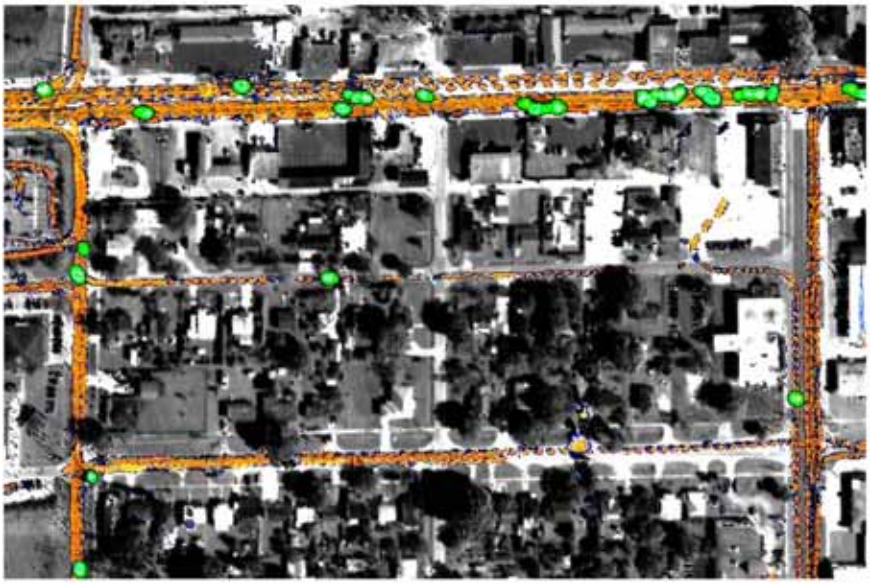

Figure 6-2: An analysis of images using dense image correspondence. The "movers" (in this case cars) are shown in green at a particular time. The orange represents the detection of the mover over 100 previous frames [7].

not been studied carefully is whether sufficient power is available. GPUs are not the only possible processing architecture although they do have advantages for image-based data. We were briefed by J. Kepner of Lincoln Labs on the use of the IBM Cell Broadband engine as a candidate for on-board processing. Further investigation of both approaches is warranted so as to understand the relative advantages and the constraints imposed by on-board power requirements [15].

The notion of using on-board processing to download the urgent data first is very much in line with the idea that when bandwidth is at a premium, filtering is essential. Preliminary analyses can be performed quickly and more detailed analyses can be performed retrospectively. Again, the use of georegistration on the compressed data will enable rapid follow up if employed in an event based architecture so that correlations with other events can be inferred rapidly.

7 GRAND CHALLENGES

The previous chapters detail important advances that make the collection, handling and basic analysis of large data feasible. As described in earlier sections, the data volumes are large but are not unmanageable based on experience with other data intensive science activities such as high energy physics, astronomy and climate research. The overarching goal in any of these activities is to understand the end requirements. There do however remain important challenges in automatic analysis of the data. Currently, human participation and intervention remains a key aspect of successful exploitation of DOD/IC data. As the data volumes grow more must be done to assist the analyst via automation. Some improvement can come from the approaches detailed in earlier sections by "triaging" data and making sure those results requiring rapid response are made available as soon as possible. In contrast, data more amenable to forensic analysis should be stored and should be easy to access using some of the architecture-neutral ideas discussed in earlier sections. However, this is only part of the solution and significant research is required to develop approaches that provide deeper analysis of data. JASON was asked by the DOD/IC whether grand challenge prize programs focused on data analysis challenges could accelerate progress in this area.

When faced with the evaluation of a scientific program and its future in this context, JASON often resorts to the notion of a "Grand Challenge". These challenges are meant to focus a field on a very difficult but imaginably achievable medium-term (ten year) goal. Via these focus areas, the community can achieve consensus on how to surmount currently limiting technological issues and can bring to bear sufficient large scale resources to overcome the hurdles. Examples of what may be viewed as successful grand challenges are the Human Genome Project, the landing of a man on the moon and, the successful navigation of an autonomous vehicle in the Mojave desert and in an urban environment.

The JASON criteria for grand challenges are

- A one decade time scale: Everything changes much too quickly for a multi-decadal challenge to be meaningful.
- Grand challenges cannot be open-ended: It is not a grand challenge to "understand the brain", because it is never quite clear when one is done. It is a grand challenge to create an autonomous vehicle that can navigate a course that is unknown in advance without crashing.
- One must be able to see one's way, albeit dimly, to a solution. When the Human Genome Project was initiated, it was fairly clear that it was in principle doable. The major issue involved speeding up of sequencing throughput and using computation (and appropriate fast algorithms) to facilitate assembly of the genome at unprecedented levels.
- Grand challenges must be expected to leave an important legacy. This criteria attempts to discriminate against one-time stunts.

With the above examples and definitions in mind, we put forth a set of suggested challenge topics that would spur further development in automated analysis of large data. It should be emphasized that our proposals below are by no means exhaustive. Instead, they are simply meant to provide example applications of a methodology that could lead to identification of such grand challenge problems and thus to a rationale for significant investment in research in the area of machine assisted analysis of large data. None of the grand challenge problems described below focus on hardware, networking or storage. There are already many provisioning challenges in these areas supported by various professional meetings such as the Supercomputing conferences. As stated previously, the challenge is not coming from infrastructure. Rather, important advances in data fusion, registration, and ultimately in machine learning are called for.

7.1 City Model Grand Challenge

The challenge is to assemble a complete image and digital elevation model (DEM) with accuracy of 1 m of a city from 3 hours of circling imagery from a UAV and 3 hours of computation. This model must support rapid enough access to support rendering of arbitrary views at human-visual speeds ($\sim$1 sec), and ray tracing by automated algorithms seeking to register new scenes against this model.

We suppose a sensor which has 100 Mpixels (e.g. 8 standard aerial imagery cameras with overlapping field of view) of a typical pixel size of 0.5 m and a 5 km field of view, and sensor metadata which include position which is accurate to 1 m, camera focal lengths and distortions which are accurate to 10^{-4} (i.e. 1 m at a range of 10 km), and orientation which is accurate to 10^{-3} (i.e. 10 m at a range of 10 km).

The aspects which need to be addressed include

- Creation of a DEM data structure which can accommodate the salient aspects of a city. Clearly an elevation drape described by posts is *not* adequate for vertical walls, but it is acceptable to ignore higher order topology such as bridges, open windows, etc. A faceted model which permits vertical walls at arbitrary location is probably called for. A triangulated irregular network (TIN) may be appropriate, although it must support the rapid lookup described above. There is a large premium in using existing data and file standards, and we suspect that the computer graphics industry has such standards already defined. The voxel based ideas detailed in Section 5.8 may also be of use here.

- Automated registration of features seen in different frames.

(a)

(b)

Figure 7-1: A Google maps rendering of the Metropolitan Museum of Art in New York. Elevation data is now available to provide 3-D perspectives

- Triangulation of common features to refine orientation accuracy of the different frames.
- Construction of the facets, edges, and vertices which make up a city.
- Computation of the images of the facets.
- Identification and deletion of change (e.g. moving cars and people), fusing the multiple views into a city which is as static as possible.

Note that most of these tasks are well within the reach of present day algorithms and computer resources, but a substantial component of the challenge is *organizational*. Data formats need to be formulated and accepted. Algorithms need to be robust enough that errors are automatically caught and corrected. Google has begun to provide some of this information (see,

for example, Figure 7-1) but this is not at the level of accuracy and completeness required for DOD/IC applications. The development of a computational infrastructure to provide this data *uniformly* to the needed DOD/IC agencies would represent a major advance. We anticipate that cross disciplinary research would be required to achieve the task.

7.2 Automated Change Detection

This challenge is a companion to the previous one. The goal is to accept a new stream of imagery from a different UAV over a city which has already been mapped, and register, frame subtract, and detect change with a latency of less than 15 minutes.

The aspects which need to be addressed include

- Automated registration of features seen in each frame, and correction of metadata.
- Assembly of a "static city" model from the incoming imagery which can be used as a reference for subtraction. Note that the different view angles and obscuration must be accounted for, presumably by reconstructing the images of the facets of the city model.
- Projection of the "static city" back into the sensor view for each frame.
- Subtraction of the "static city" from each frame, ideally with registration refinement and intensity and point spread function matching.
- Detection of changes arising from motion of vehicles, people, etc., lights changing their state, or other activity.
- A report of change which includes position (long, lat, elevation) and uncertainties, flux change and uncertainty, shape change, and the same

Figure 7-2: Satellite imagery of Cheltenham, UK showing the central location of the Government Headquarters (GCHQ). The goal of the geolocation grand challenge is to take unlabeled imagery and determine the location.

properties derived from the "static city" if an object is present there as well.

7.3 Geolocation Grand Challenge

The goal of this grand challenge is to develop methodologies to geolocate imagery that comes with no location metadata. Given some imagery to be provided to the challenge team the goal would be to develop a DEM model and data architecture that could correlate the scenery (adjusting for the random perspective) and identify a location. This could have several levels of difficulty. For example, one might start with aerial imagery of the type shown in Figure 7-2 to determine location.

A much more difficult problem would be to take video camera data taken at ground level and attempt to geolocate the scenery. This would require a 3-D model of the earth that would track changes over time to scenery.

Efficient algorithms for image correlation would need to be developed to identify the location.

7.4 Conversational Analysis Grand Challenge

So far we have focused solely on grand challenges associated with image analysis. It is also important to consider analysis of other media such as voice or text. Some of the algorithms presented in Section 5 are useful for identifying words in a document or in an audio transmission but this is far from inferring the context of a conversation of interest. A grand challenge in this area would bring the machine learning community together to assess and improve the state of the art in this area. In order to provide a data source that all entrants can access, we would propose to provide an open source corpus of calls into a radio talk show. The challenge would be to analyze some aspects of the context of the conversation. For example, can one determine whether a given caller is in support of or is opposed to the views of the host. Another challenge would be to summarize the conversation. This could be extended to perform the same type of analysis with both video and audio. Challenges might be to identify the speakers and to infer their positions on issues. This is currently of interest to information providers such as Google. They are currently analyzing the audio stream from news videos and tagging the content by using speech to text translation. This allows one to search news casts for particular topics and is particularly important to an overall search capability. Google has made this publicly available as a "Google gadget" shown in Figure 7-3.

Figure 7-3: A Google gadget that analyzes the audio portion of video news so that it can be searched for specific tags.

7.5 Role Discovery Grand Challenge

The objective of this grand challenge is to infer the membership and roles of an organization of interest through an analysis of their communications.

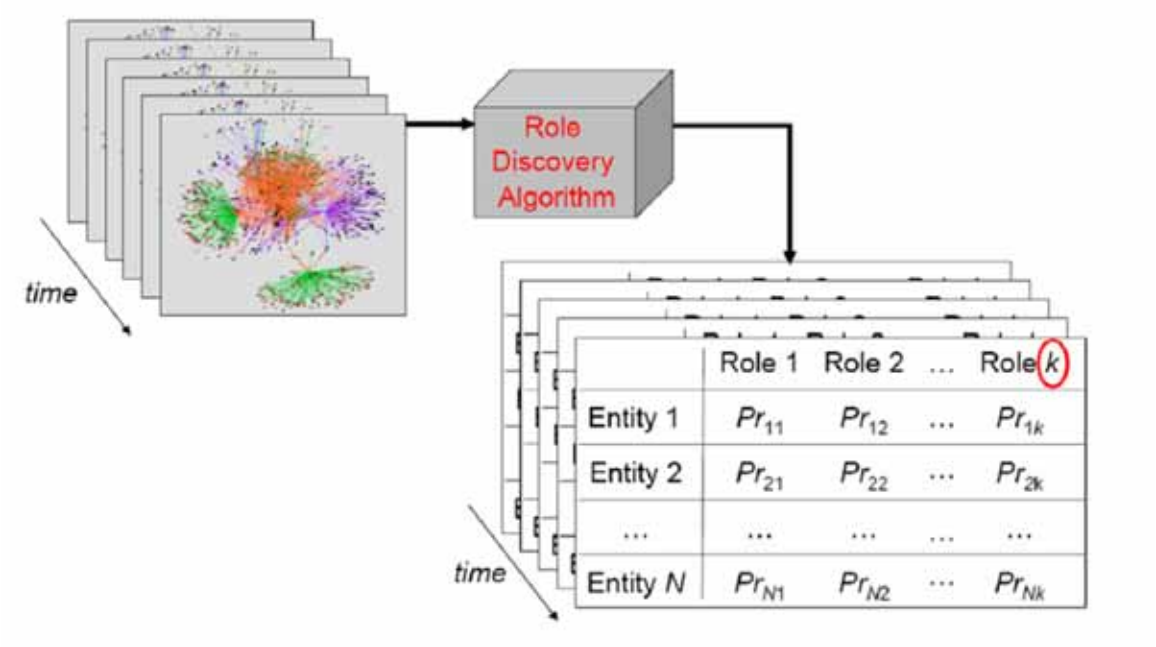

Figure 7-4: The use of network analyses and machine learning on a corpus of open source e-mails to infer role discovery [11]

.

This is a key part of social network analysis. In order to allow open access, a corpus of publicly available documents would have to be developed or one might contemplate generating synthetic data. An example is the collection of Enron e-mails which were made public after the collapse of the company. The use of machine learning techniques to infer roles is an active research area. A graphical representation of what is required is shown in Figure 7-4. We were briefed by Tina Eliassi-Rad of LLNL on the application of network analysis tools on the Enron corpus to determine the role of various employees. An additional objective would be to characterize how roles evolve over time [11].

7.6 Cross Disciplinary Collaborative Challenge

The idea of this grand challenge is to support the development of ba-

sic research in machine learning and information science. We consider the teaming together some of top bio-informaticians with the top intelligence-informaticians to work a grand challenge in personalized medicine. Today, medicine is being challenged with how to meet the future of personalizing treatment while lowering the cost of services. The belief is that the creation of and access to electronic medical records and research records can change the face of health, wellness, and medical practice. The intelligence field is faced with the similar daunting task. JASON sees a range of experts from the intelligence areas striving to develop methods of knitting together all forms of information to build coherent assessments of various conditions.

Bringing together the strengths of these disciplinary-similar communities, and possible unique approaches to problems, could help directly contribute to the vision of personalized medicine. For example, new imaging technologies are being developed daily, new understandings of disease progressions are being discovered, advances in nanoscience and technology are opening up avenues for drug delivery. There are also aggressive activities that have been mounted to begin to capture and manage the information, e.g. semantic web. There is, however, a gap emerging in where we are today to where we need to go to achieve the personalized medicine vision. This gap is in the area of integrated informatics (of all sorts). Major emphasis must be placed on the integration of what can be learned from the massive amount of information generated and the creation of actionable knowledge. It is through the integration and use of the information that medicine and medical treatment will be transformed.

The feedback loop of this collaboration should result in new methods for the intelligence community. The reason for mounting the exercise in the personalized medicine area is to keep access as open as possible and to draw a large range of interest. There may even be an opportunity for collaboration between DOD and NIH.

The challenge would need to pose some complex questions regarding personalized health, wellness, and the delivery of services. The challenge would need to identify a starting set of data sources. If the challenge focused on cancer, several data sources being created by the National Cancer Institute would be leveraged. These could include Cancer Bioinformatics Grid (caBIG) [14], The Cancer Genome Atlas [18], and anonymized patient hospital or clinical records perhaps through the Veterans Administration. Other sources of data may be identified by the collaborative team(s).

If the challenge were to go beyond modeling, into how the information gets conveyed to the health care providers, it would benefit the collaboration to include expertise from social behavioral sciences and management sciences.

8 CONCLUSION

The preceding sections provide some context for the large data problems faced by the DOD. Comparisons with activities in data intensive science areas such as high energy physics and astronomy show that the data volume for all these activities is certainly challenging (hundreds of Petabytes) but, as has been seen, this is not an unmanageable data volume. Significant filtering of the data is a key component of any data collection activity. Sometimes this has to be done at the data source and in other cases can be done retrospectively. In all such cases, an understanding of the end requirements is the best way to assess the relevant data size and the corresponding required infrastructure. As has been seen for data sets on the order of hundreds of Petabytes, data storage technology will keep up even in the face of flattening technology trends for single storage devices.

Requirements for the handling of data (particularly wide area surveillance data) will differ depending on timeliness requirements. Where time permits detailed retrospective analysis, JASON recommends the use of homogeneous data architectures, "cloud computing" (the provisioning of services from a generic cloud of servers) and the use of streaming data analysis algorithms that do not tie the data to particular data base schema or to a specific set of queries. Such approaches are currently in wide use by information providers such as Google and others. One could envision several facilities that can share resources to provide an overall capability that can be shared among multiple agencies. There are issues of security of course and these are quite complex. It is obviously critical to preserve aspects of security classification and also to ensure that a proper "need to know" is verified before some piece of data is incorporated in a specific analysis. However, the potential advantages that arise from seamless data sharing among appropriate DOD and IC agencies could be significant.

On more intermediate time scales, a service oriented architecture is ap-

propriate and such applications are being deployed by the DOD/IC. This approach is also connected to the use of homogeneous data architectures. The web services would utilize the data store and mine it using approaches perhaps along the same lines as those described in Section 5 although clearly more work is required in this area that is specific to DOD/IC objectives. Once mined, services could be integrated to form web based applications that could be used to fuse diverse data and present it in the most informative manner to the relevant analyst or decision maker.

When rapid response is required, a push-based or event-driven architecture is most appropriate. For DOD/IC applications the most critical metadata is accurate space and time registration. Combined with more accurate georegistration capabilities this will more easily facilitate the analysis of correlated activity in locations of interest. Again, the event streams can be monitored in real time or stored and mined later for correlations.

The key issue is not the availability or development of hardware; there seems to be ample capability in this regard both in the development of data sources (sensors) and data storage media. What does seem to be lacking is an adequate investment in software, so that the analyst can keep pace with the impressive developments to date in wide area surveillance.

As the greatest challenge will come from the need to automate analysis, the most immediate need is for algorithmic advances that can help cue the analyst and trigger closer observation as well as possible fusing of other relevant data. The notion of fully automated analysis is today at best a distant reality, and for this reason it is critical to also invest in research to promote algorithmic advances; one way to effectively engage the relevant research communities is through the use of grand challenges in the area of data analysis and machine learning. The key requirements for such grand challenges are that they focus on a difficult but ultimately achievable goal, be science-driven, and that success will leave a clear legacy in the target area. Several such challenges have been suggested but it would be useful to con-

sider other challenges solicited from broader communities that could engage the research community.

Our findings as regards data analysis challenges for the DOD/IC are as follows:

- DOD/IC data volumes as generated via various sensing modalities are, and will continue to be, significant, but they are in many ways comparable to those faced by other large enterprises.
- Important parallels can be drawn with data intensive science efforts such high energy physics and astronomy.
- End user analysis requirements must drive the design of all aspects of the data enterprise including storage, database design and analysis tools.
- At present there is insufficient investment in software to more effectively process data as opposed to hardware to both collect and store data.
- Data organization and processing approaches such as cloud computing would appear to be best suited at present to facilitate future data fusion and discovery.
- Continued investment in technologies such as service-oriented architecture coupled with additional investment in event-driven architecture and software will be of benefit in enabling data fusion across the DOD/IC enterprise.
- Significant gains in data fusion can be realized in the short term through accurate spatial georegistration and time registration of sensor data,
- Processing closer to the sensor can yield important benefits provided there is a clear formulation of critical time sensitive data requirements.
- The greatest challenge will come from the need to perform automated analysis in support of the DOD/IC analyst.

- Grand challenges to stimulate further research in automated analysis can be used to assess and prioritize future research activities.

Given these findings, JASON recommends as follows:

- The DOD/IC communities should formulate a data analysis doctrine that
  - Continually assesses data requirements by matching analysis objectives to the data stream,
  - Focuses on homogeneous storage solutions with open interfaces,
  - Focuses on flexible analytic techniques that do not tie data to the query,
  - Focuses as strongly on software development as it does on sensor, storage, and network development,
  - Differentiates between time sensitive analyses and retrospective analyses and applied the appropriate paradigm in each case.
- The DOD/IC communities should put into place efforts to validate the doctrine via several use cases.
- Continued investment should take place in interdisciplinary research in data analytics, machine learning and optimization.
- Invest in several grand challenges to assess and improve the state of the art in automated data analysis.

A APPENDIX: Briefers

Briefer	Affiliation	Briefing title
Gordon Ainsworth	NGA	Activity Model
Dr. Chris Arney	Army Research Office	Data Analysis Challenges & Shifting the Decision-Making Paradigm
Dr. Jim Ballas	Naval Research Lab	Using Web Services for Streaming Content
Dr. Rich Baraniuk	Rice University	Compressive sensing
Dr. Jacek Becla	SLAC	Data Intensive Data Management
Dr. Amy Braverman	JPL	Massive Data Set Analysis in Climate Research at JPL
Dr. Nevin Bryant	JPL	Path to Automatic and Precise Global Imagery Co-registration
Dr. Julian Bunn	Caltech	Tera, Peta, and Exabyte data collection distribution, analysis and management for CERN's large hadron collider
Dr. Randal Burns	Johns Hopkins University	The store everything model for high performance computing
Dr. John Callahan	APL, Johns Hopkins	IARPA Blackbook and RDEC
Dr. Mark Duchaineau	Lawrence Livermore National Lab	Sensor-based video processing
Dr. Georg Djorgovski	Caltech	Real-Time Mining, Anomalous Event Detection, and Follow-Up in massive Data Streams: Examples and Challenges from Synoptic Sky Surveys
Dr. Tina Eliassi-Rad	Lawrence Livermore National Lab	Role discovery in dynamic graphs
Dr. Eric Fetzer	JPL	Challenges to understanding Earth's climates with Satellite Observations
Dr. Maya Gokhale	Lawrence Livermore National Lab	Data-Intensive Supercomputing Architectures
Dr. Roger Haskin	IBM Almaden	Storage for Data Intensive Computing
Dr. Bobby Junker	Office of Naval Research	Fusion / Integration of Large Databases of Disparate Information
Dr. Jeremy Kepner	Lincoln Lab	Processing closer to the sensor data; implementation challenges
Dr. Scott Kohn	Lawrence Livermore National Lab	Document Triage via Faceted Search and Architectures for Large Semantic Graphs
Martin Kruger	Office of Naval Research	Human Terrain Research
Dr. Mark Linderman	AFRL, Rome	Information management, Experimental results
Dr. Steven Low	Caltech	Internet Congestion Control
Dr. John Marion	Logos Technologies	Constant Hawk
Dr. Jim Siegrist	Lawrence Berkeley Lab	Data Handling and Data Fusion Issues in High Energy Physics: lessons for DOD
Dr. Alex Szalay	Johns Hopkins University	Petascale Scalable Computing
Dr. John Tonry	University of Hawaii and JASON	Pan-STARRS - Hardware and Pipeline Analysis Challenges
Dr. Craig Tull	Lawrence Berkeley Lab	The Software Framework Approach to HEP Data Handling
Dr. Jeffrey Ullman	Stanford University	My favorite algorithms for large scale similarity search
Dr. Cliff Weinstein	Lincoln Laboratory	Research in Modeling, Simulation and Recognition of Terror Networks and Threat Scenarios

References

[1] Dave Anderson, Jim Dykes, and Erik Riedel. More than an interface–SCSI vs. ATA. In *Proceedings of the Second USENIX Conference on File and Storage Technologies (FAST)*, San Francisco, CA, March 2003.

[2] James Ballas. Using web services framework for streaming content. Presentation to JASON, June 24, 2008.

[3] Jacek Becla. Data-intensive data management. Presentation to JASON, June 29, 2008.

[4] Peter J. Braam. The Lustre storage architecture. http://www.lustre.org/documentation.html, Cluster File Systems, Inc., August 2004.

[5] Julian Bunn. Tera-, peta- and exabyte data collection, distribution, analysis and management for cern's large hadron collider. Presentation to JASON, June 25, 2008.

[6] Jack Callahan. Blackbook2 and RDEC. Presentation to JASON, June 25, 2008.

[7] Mark Duchaineau. Sensor based video processing. Presentation to JASON, June 23, 2008.

[8] Jon G. Elerath. Specifying reliability in the disk drive industry: No more MTBF's. In *Proceedings of 2000 Annual Reliability and Maintainability Symposium*, pages 194–199. IEEE, 2000.

[9] Jon G. Elerath and Michael Pecht. Enhanced reliability modeling of RAID storage systems. In *Proceedings of the 2007 Int'l Conference on Dependable Systems and Networking (DSN 2007)*, pages 175–184. IEEE, June 2007.

[10] Jon G. Elerath and Sandeep Shah. Server class disk drives: How reliable are they? In *Proceedings of 2004 Annual Reliability and Maintainability Symposium*, pages 151–156. IEEE, 2004.

[11] Tina Eliassi-Rad. Machine learning on graphs. Presentation to JASON, June 23, 2008.

[12] Bob Gourley. Thoughts on the future of information sharing technology, 2008. http://ctovision.typepad.com/ InfoSharingTechnologyFutures.ppt.

[13] Roger Haskin. Storage for data intensive computing. Presentation to JASON, June 26, 2008.

[14] National Cancer Institute. Cancer bio-informatics grid. `http://cabig.nci.nih.gov`.

[15] Jeremy Kepner. Processing closer to the sensor data - implementation challenges. Presentation to JASON, June 23, 2008.

[16] David Luckham. Thoughts on the future of information sharing technology. `http://complexevents.com`.

[17] Andrei Broder and Michael Mitzenmacher. Network applications of bloom filters: A survey. In *Internet Mathematics*, pages 636–646, 2002.

[18] National Institutes of Health. The cancer genome atlas (tcga). `http://cancergenome.nih.gov/`.

[19] PanSTARRS. Thoughts on the future of information sharing technology. `http://pan-starrs.ifa.hawaii.edu/public/`.

[20] T. Pollard and J.L. Mundy. Change detection in a 3-d world. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–6, June 2007.

[21] Atlas project. Atlas project web site. `http://atlas.ch`.

[22] Frank Schmuck and Roger Haskin. GPFS: A shared-disk file system for large computing clusters. In *Proceedings of the 2002 Conference on File and Storage Technologies (FAST)*, pages 231–244. USENIX, January 2002.

[23] Jim Siegrist. Data and storage framework infrastructure for high energy physics. Presentation to JASON, June 25, 2008.

[24] InPhase technologies. Holographic storage. `http://www.inphase-technologies.com`.

[25] John Tonry. Pan-starrs - distilling science from petabytes. Presentation to JASON, June 26, 2008.

[26] Jeffrey Ullman. My favorite algorithms for large scale data mining. Presentation to JASON, June 27, 2008.

[27] Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long, and Carlos Maltzahn. Ceph: A scalable, high-performance distributed file system. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI)*, Seattle, WA, November 2006. USENIX.

[28] Brent Welch, Marc Unangst, Zainul Abbasi, Garth Gibson, Brian Mueller, Jason Small, Jim Zelenka, and Bin Zhou. Scalable performance of the Panasas parallel file system. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST)*, pages 17–33, February 2008.

[29] Qin Xin, Ethan L. Miller, Thomas J. E. Schwarz, and Darrell D. E. Long. Impact of failure on interconnection networks in large storage systems. In *Proceedings of the 22nd IEEE / 13th NASA Goddard Conference on Mass Storage Systems and Technologies*, Monterey, CA, April 2005.

DISTRIBUTION LIST

Administrator
U.S. Dept of Energy
National Nuclear Security Administration
1000 Independence Avenue, SW
NA-10 FORS Bldg
Washington, DC 20585

Assistant Secretary of the Navy
(Research, Development & Acquisition)
1000 Navy Pentagon
Washington, DC 20350-1000

Assistant Deputy Administrator for
Military Application
NA-12
National Nuclear Security Administration
U.S. Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

DARPA Library
3701 North Fairfax Drive
Arlington, VA 22203-1714

Defense Technical Information Center (DTIC)
8725 John J. Kingman Road
ATTN: DTIC-OA (Mr. Jack Rike)
Suite 0944
Fort Belvoir, VA 22060-6218

Deputy Under Secretary of
Defense Science & Technology
3040 Defense Pentagon
Washington, DC 20301-3040

Deputy Chief Scientist
U.S. Army Space & Missile Defense Command
PO Box 15280
Arlington, VA 22215-0280

Director, IDA
Technical Information Services
Room 8701
4850 Mark Center Drive
Alexandria, VA 22311-1882

Director, IARPA
7005 52nd Avenue
College Park, MD 20742

Director, DTRA
Research Development Office
8725 John Jay Kingman Road
Room 3380, Mail Stop 6201
Fort Belvoir, VA 22060

Director of Space and SDI Programs
SAF/AQSC
1060 Air Force Pentagon
Washington, DC 20330-1060

Headquarters Air Force XON
4A870 1480 Air Force Pentagon
Washington, DC 20330-1480

IC JASON Program [2]
Chief Technical Officer/OCS
2P0104 NHB
Central Intelligence Agency
Washington, DC 20505-0001

JASON Library [5]
The MITRE Corporation
3550 General Atomics Court
Building 29
San Diego, CA 92121-1122

Records Resource
The MITRE Corporation
Mail Stop C025
202 Burlington Road, Rte 62
Bedford, MA 01730-1420

Principal Deputy Director
Office of Science, SC-2/Forrestal Building
U.S. Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

Reports Collection
Los Alamos National Laboratory
Mail Station 5000
MS A150
PO Box 1663
Los Alamos, NM 87545

Superintendent
Code 1424
Attn: Documents Librarian
Naval Postgraduate School
Monterey, CA 93943

U S Army Space & Missile Defense Command
Attn: SMDC-ZD (Dr. Swinson)
PO Box 1500
Huntsville, AL 35807-38017

Dr. Lawrence K. Gershwin
NIC/NIO/S&T
2E42, OHB
Washington, DC 20505

Dr. Alfred Grasso
President & CEO
The MITRE Corporation
Mail Stop N640
7515 Colshire Drive
McLean, VA 22102-7508

Dr. Barry Hannah
Reentry Systems Branch Head, Navy Strategic Systems Programs
Strategic Systems Programs (Attn: SP28)
2521 Clark Street, Suite 1000
Arlington, VA 22202-3930

Dr. Robert G. Henderson
The MITRE Corporation
Mailstop MDA/ Rm 5H305
7515 Colshire Drive
McLean, VA 22102-7508

Dr. Bobby R. Junker
Office of Naval Research
Code 31
800 North Quincy Street
Arlington, VA 22217-5660

Mr. Kevin "Spanky" Kirsch
Director, Special Programs
US Department of Homeland Security
Science and Technology Directorate
Washington, DC 20528

Dr. Daniel J. McMorrow
Director, JASON Program Office
MITRE Corporation
Mailstop T130
7515 Colshire Drive
McLean, VA 22102-7508

Dr. Julian C. Nall
Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311-1882

Dr. John R. Phillips
Chief Scientist, DST/CS
2P0104 NHB
Central Intelligence Agency
Washington, DC 20505-0001

Dr. William S. Rees, Jr.
OSD/DDR&E
Deputy Under Secretary of Defense for Laboratories and Basic Sciences
3030 Defense Pentagon
Room 3C913A
Washington, DC 20301-3030

Dr. Scott P. Sarlin
Director S&T (Acting)
Room 5B318 LX-2
Washington, DC 20515

Mr. Alan R. Shaffer
Principal Deputy Director
DDR&E
3040 Defense Pentagon, Room 3B 854
Washington, DC 20301-3040

Mr. Anthony J. Tether
DIRO/DARPA
3701 N. Fairfax Drive
Arlington, VA 22203-1714